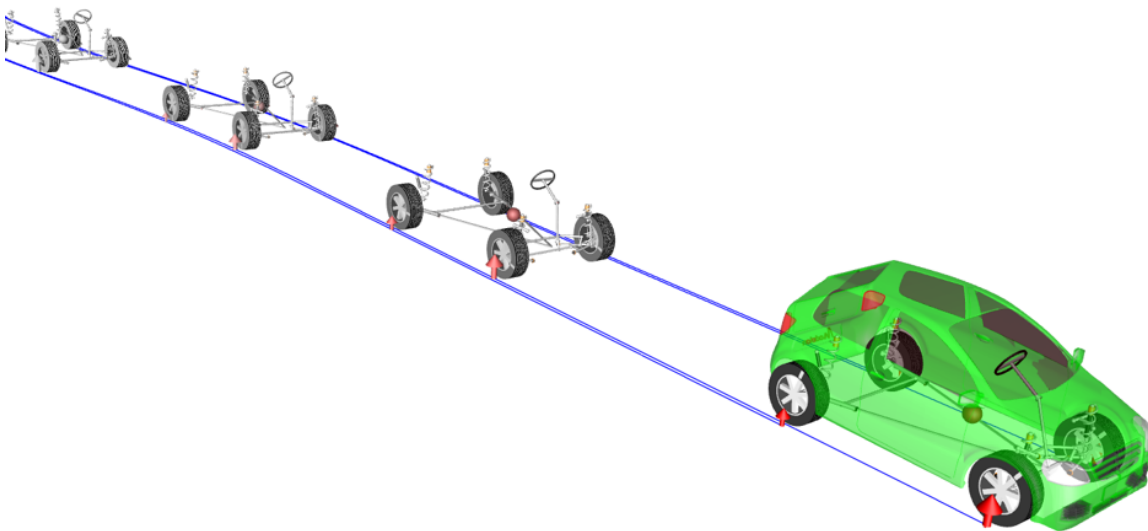




# CHALMERS

---



## Implementation and Evaluation of a Flexible Beam Model in Vehicle Dynamics Applications

Master's thesis in Automotive Engineering

SIDHARTH MALIK



MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

# Implementation and Evaluation of a Flexible Beam Model in Vehicle Dynamics Applications

SIDHARTH MALIK

Department of Applied Mechanics  
Division of Dynamics  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden 2016

Implementation and Evaluation of a Flexible Beam Model in Vehicle Dynamics Applications  
SIDHARTH MALIK

© SIDHARTH MALIK, 2016

Master's thesis 2016:34  
ISSN 1652-8557  
Department of Applied Mechanics  
Division of Dynamics  
Chalmers University of Technology  
SE-412 96 Göteborg  
Sweden  
Telephone: +46 (0)31-772 1000

Cover:

Open loop steer, double lane change experiment of a compact car chassis using compliant double wishbone front and twist beam rear suspension in VDL

Chalmers Reproservice  
Göteborg, Sweden 2016

Implementation and Evaluation of a Flexible Beam Model in Vehicle Dynamics Applications  
Master's thesis in Automotive Engineering  
SIDHARTH MALIK  
Department of Applied Mechanics  
Division of Dynamics  
Chalmers University of Technology

## ABSTRACT

This thesis was done together with Modelon AB who develop and maintain the Vehicle Dynamics Library (VDL) for Dymola which is a graphical modeling and simulation environment based on the Modelica language. VDL currently supports detailed vehicle models built using rigid multibody components. However, there is a need to include component flexibility in these models to increase the fidelity. Recently, a flexible beam model was implemented in Dymola as part of a Master's thesis using the Euler-Bernoulli theory in a floating frame of reference formulation. As part of the present thesis, a particular component has been identified in vehicle suspensions (twist beam) and modeled using the flexible beam. This suspension assembly has been validated against and shows good agreement with an ADAMS/Car model of comparable fidelity. The twist beam rear suspension has also been implemented in a chassis model and simulated in real time in Dymola using a fixed step solver.

Keywords: Dymola, Vehicle Dynamics, Twist Beam, Multi Body Dynamics



## PREFACE

This thesis was done as a partial requirement for the fulfillment of MSc Automotive Engineering degree at Chalmers University of Technology. It was carried out together with Modelon AB and the Division of Dynamics, Department of Applied Mechanics at Chalmers from January to June 2016. This thesis was carried out with Peter Folkow, Associate Professor and Head of Division of Dynamics as examiner from Chalmers, Anders Ericsson and Peter Sundström as supervisors from Modelon AB. Additional inputs from Bengt Jacobson, Professor and group leader Vehicle Dynamics, Division of Vehicle Engineering and Autonomous Systems, Chalmers regarding applications of the beam model are highly appreciated.

Göteborg, Sweden, June 2016  
Sidharth Malik



## NOMENCLATURE

|      |                                                |
|------|------------------------------------------------|
| ACAR | ADAMS/Car                                      |
| DAE  | Differential Algebraic Equation                |
| DLR  | Deutsches Zentrum für Luft- und Raumfahrt e.V. |
| FFoR | Floating Frame of Reference                    |
| FMI  | Functional Mock-up Interface                   |
| MBS  | Multibody simulation                           |
| MNF  | Modal Neutral File                             |
| NVH  | Noise, Vibration and harshness                 |
| VDL  | Vehicle Dynamics Library                       |



# CONTENTS

|                                                             |            |
|-------------------------------------------------------------|------------|
| <b>Abstract</b>                                             | <b>i</b>   |
| <b>Preface</b>                                              | <b>iii</b> |
| <b>Nomenclature</b>                                         | <b>v</b>   |
| <b>Contents</b>                                             | <b>vii</b> |
| <b>1 Introduction</b>                                       | <b>1</b>   |
| 1.1 Background . . . . .                                    | 1          |
| 1.2 Limitations . . . . .                                   | 1          |
| 1.3 Previous work . . . . .                                 | 1          |
| 1.4 Dymola environment . . . . .                            | 1          |
| 1.5 Vehicle Dynamics Library . . . . .                      | 2          |
| <b>2 Theory</b>                                             | <b>3</b>   |
| 2.1 The Floating Frame of Reference Formulation . . . . .   | 3          |
| 2.2 Modal Flexibility . . . . .                             | 3          |
| 2.3 Implementation in Dymola . . . . .                      | 4          |
| 2.3.1 Model structure . . . . .                             | 4          |
| 2.3.2 FlexBeam component . . . . .                          | 5          |
| 2.3.3 FlexBeamME component . . . . .                        | 6          |
| 2.3.4 FlexBeamBushings component . . . . .                  | 6          |
| 2.4 State selection in VDL . . . . .                        | 6          |
| 2.5 Twist Beam Suspension . . . . .                         | 7          |
| 2.5.1 Current state of the art . . . . .                    | 8          |
| 2.5.2 Implementation in VDL . . . . .                       | 8          |
| 2.6 Output variables of suspension tests . . . . .          | 9          |
| <b>3 Method</b>                                             | <b>10</b>  |
| 3.1 Twist Beam Modeling in ADAMS/Car . . . . .              | 10         |
| 3.1.1 Coordinate system . . . . .                           | 10         |
| 3.1.2 Template modification . . . . .                       | 10         |
| 3.1.3 Hardpoints . . . . .                                  | 11         |
| 3.1.4 Spring . . . . .                                      | 12         |
| 3.1.5 Damping . . . . .                                     | 13         |
| 3.1.6 Bushing properties . . . . .                          | 14         |
| 3.1.7 Suspension assembly and test rig . . . . .            | 15         |
| 3.1.8 Tire properties . . . . .                             | 16         |
| 3.2 Beam Modeling in ABAQUS . . . . .                       | 17         |
| 3.2.1 Beam cross section . . . . .                          | 17         |
| 3.2.2 Material properties . . . . .                         | 17         |
| 3.2.3 Beam element type and orientation . . . . .           | 18         |
| 3.2.4 Modal reduction in ABAQUS/CAE . . . . .               | 19         |
| 3.3 Twist Beam Modeling in VDL . . . . .                    | 19         |
| 3.3.1 Coordinate system . . . . .                           | 19         |
| 3.3.2 Linkage modification . . . . .                        | 20         |
| 3.3.3 Beam model . . . . .                                  | 21         |
| 3.3.4 Bushings in VDL . . . . .                             | 21         |
| 3.3.5 Suspension Assembly . . . . .                         | 21         |
| 3.3.6 Test rigs . . . . .                                   | 22         |
| 3.3.7 Importing ADAMS/Car data into VDL . . . . .           | 23         |
| 3.3.8 State selection and performance improvement . . . . . | 24         |
| 3.4 Cantilever beam test model . . . . .                    | 27         |

|          |                                                                                              |           |
|----------|----------------------------------------------------------------------------------------------|-----------|
| 3.5      | Kinematics and Compliance tests in VDL and ADAMS/Car . . . . .                               | 29        |
| 3.6      | Chassis Modeling in VDL . . . . .                                                            | 30        |
| 3.6.1    | Real time simulation in Dymola . . . . .                                                     | 31        |
| 3.6.2    | Dymola decouple block . . . . .                                                              | 31        |
| 3.6.3    | Chassis experiments . . . . .                                                                | 33        |
| 3.6.4    | Chassis model variants tested for real time performance improvement . . . . .                | 34        |
| <b>4</b> | <b>Results</b>                                                                               | <b>35</b> |
| 4.1      | Component Level Tests on Cantilever Beam . . . . .                                           | 35        |
| 4.2      | Performance Evaluation and State Selection . . . . .                                         | 35        |
| 4.2.1    | Effect of parameter evaluation . . . . .                                                     | 35        |
| 4.2.2    | Effect of different FlexBeam components . . . . .                                            | 36        |
| 4.2.3    | Effect of turning on dynamic modes in the FlexBeam component . . . . .                       | 36        |
| 4.3      | Suspension Kinematics Analysis . . . . .                                                     | 39        |
| 4.3.1    | Parallel wheel travel . . . . .                                                              | 39        |
| 4.3.2    | Opposite wheel travel . . . . .                                                              | 41        |
| 4.3.3    | Roll motion . . . . .                                                                        | 43        |
| 4.4      | Suspension Compliance Analysis . . . . .                                                     | 45        |
| 4.4.1    | Lateral force compliance . . . . .                                                           | 45        |
| 4.4.2    | Longitudinal force compliance . . . . .                                                      | 46        |
| 4.4.3    | Aligning torque compliance . . . . .                                                         | 47        |
| 4.5      | Chassis Model . . . . .                                                                      | 48        |
| 4.5.1    | Open loop steering test . . . . .                                                            | 48        |
| <b>5</b> | <b>Conclusion</b>                                                                            | <b>53</b> |
| 5.1      | Implementation in VDL . . . . .                                                              | 53        |
| 5.2      | ADAMS/Car Benchmarking . . . . .                                                             | 53        |
| 5.3      | Chassis Experiments in VDL . . . . .                                                         | 53        |
| <b>6</b> | <b>Future Work</b>                                                                           | <b>54</b> |
| 6.1      | Beam Model . . . . .                                                                         | 54        |
| 6.2      | Twist Beam Suspension Model . . . . .                                                        | 54        |
| 6.3      | Additional Applications of the Beam Model . . . . .                                          | 54        |
| 6.4      | VDL Usage in Vehicle Dynamics CAE . . . . .                                                  | 54        |
|          | <b>References</b>                                                                            | <b>55</b> |
|          | <b>Appendices</b>                                                                            | <b>57</b> |
| <b>A</b> | <b>ABAQUS Input Decks</b>                                                                    | <b>58</b> |
| A.1      | ABAQUS Input deck for unit test beam MNF generation(*.inp format) . . . . .                  | 58        |
| A.2      | ABAQUS Input deck for twist beam MNF generation(*.inp format) . . . . .                      | 59        |
| <b>B</b> | <b>ABAQUS Output</b>                                                                         | <b>60</b> |
| B.1      | Partial ABAQUS output for substructure generation(*.dat format) . . . . .                    | 60        |
| <b>C</b> | <b>ADAMS/Car files</b>                                                                       | <b>61</b> |
| C.1      | ADAMS/Car assembly file(TeimOrbit format) . . . . .                                          | 61        |
| C.2      | ADAMS/Car subsystem file(TeimOrbit format) . . . . .                                         | 63        |
| <b>D</b> | <b>Dymola translation and simulation logs</b>                                                | <b>69</b> |
| D.1      | Example translation log from a VDL suspension experiment . . . . .                           | 69        |
| D.2      | Example simulation log from a VDL suspension experiment . . . . .                            | 70        |
| D.3      | Partial translation log from a VDL suspension experiment running on multiple cores . . . . . | 71        |

# 1 Introduction

## 1.1 Background

This thesis was done together with Modelon AB which is a Swedish company specialized in model based systems engineering using the Modelica and Functional Mock-up Interface(FMI) standards.

Modelica [24] is a free object oriented modeling language for component oriented modeling of dynamic systems. Dymola [11] is a commercial modeling and simulation environment distributed by Dassault Systemes AB which utilizes the free Modelica libraries as well as commercial libraries for specialized domains. Modelon AB maintains and develops multiple Modelica libraries, one of which is the Vehicle Dynamics Library (VDL) [26]. VDL enables the modeling of conceptual as well as detailed multi body vehicle models and performance evaluation at the subsystem and vehicle level. As VDL is based on the Modelon Base Library (itself an extension of the Modelica Standard Library), at present there is no way to include component flexibility in VDL models. Even though a commercial DLR (Deutsches Zentrum für Luft- und Raumfahrt e.V.) flexible bodies library is available [14], this thesis builds on work done earlier at Modelon to develop an in-house library to model flexible beams [17].

Two specific components were initially identified: a twist beam and an anti-roll bar, for the implementation of the beam model based on the effect of flexibility on axle as well as full vehicle behavior. For validation specific to vehicle dynamics performance evaluation, a commercial tool, ADAMS/Car was used which is a multibody modeling and simulation environment by MSC Software [32]. Due to time constraint, only the twist beam was modeled and validated as part of this thesis.

## 1.2 Limitations

The following simplifications and assumptions have been used in this thesis:

- The beam model is an Euler-Bernoulli beam valid for small deformations.
- Modal reduction is used for calculating deformation of the beam.
- Cross section offset effects have been neglected.
- The warping effects of the cross beam to trailing arm connection have been neglected.
- Only the cross beam is flexible as part of this thesis, the trailing arm is considered to be rigid.
- Only the chassis mount connection is using a bushing and the rest of the connections are using ideal joints.

## 1.3 Previous work

Since the initiation of the flexible bodies project at Modelon, three master theses have been performed, all but one of them utilizing the Floating Frame of Reference (FFoR) formulation in combination with other beam theories. One of them was based on the finite element formulation of flexible bodies.

The latest of these was performed in 2015 [17], using the Euler Bernoulli beam theory along with FFoR. The beam models have been shown to have good correlation to analytic calculations as well as a finite element beam modeled in ABAQUS for static and dynamic loads. The beam model has also been used as part of kinematic loops in Dymola(slider crank).

## 1.4 Dymola environment

The Dymola environment consists of two self explanatory model editing and simulation modes:

- Modeling
- Simulation

The modeling mode has three layers enabling different aspects of modeling and model management:

- Diagram layer: For model editing using components and connection.
- Text layer: For editing Modelica text.
- Icon layer: For editing display icon of the active model.

The majority of modeling work done in this thesis was done in the diagram layer.

## 1.5 Vehicle Dynamics Library

The Vehicle Dynamics Library provides a comprehensive library of reusable components and subsystems which can be used to build suspension assemblies and full vehicle models. The development of this library in its current form is well documented in [2, 3, 5, 16]. The Modelica language specifications and the Dymola environment allow scaling of model detail and drag and drop nature of components in the VDL. Advanced users can also use the Modelica code and library support to build their own models. Recent developments in Dymola and VDL enable the parallelization of detailed multibody vehicle models having 150 – 300 degrees of freedom and simulation in real time [4]. Data access components allow the usage of model parameters from multiple modeling environments to be used in the VDL model as shown in [4]. A similar approach was used for ADAMS/Car data as part of this thesis.

Additionally, VDL contains test rigs for suspension systems and full vehicles along with driver and road models which enable experimentation and validation of these models.

A twist beam implementation is available in the **TwistBeamTT** model utilizing a bushing element which provides the necessary compliance between the left and right wheels as shown in Figure 1.1. This model was modified to use the beam model instead.

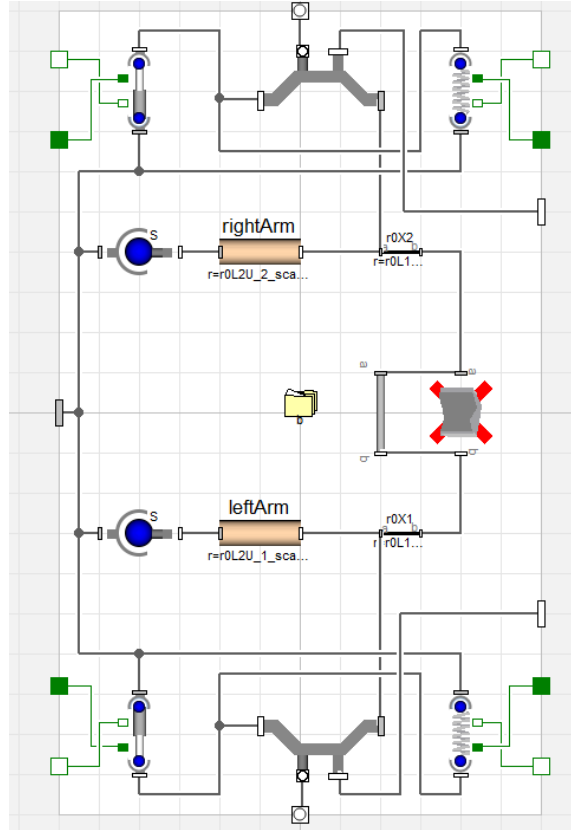


Figure 1.1: *Linkage model of TwistBeamTT model in VDL.*

## 2 Theory

### 2.1 The Floating Frame of Reference Formulation

The floating frame of reference formulation is a method to describe a flexible body in terms of two sets of coordinates: *reference* and *elastic* coordinates [35]. Reference coordinates are used to define the location and orientation of body reference while the elastic coordinates describe body deformation with respect to the reference coordinate.

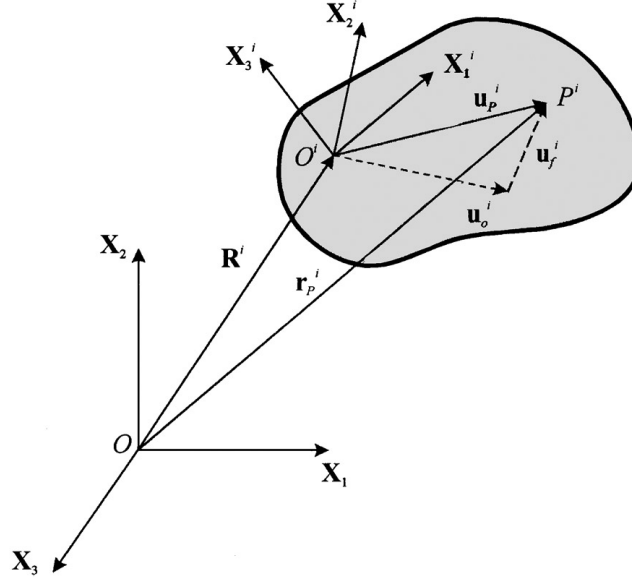


Figure 2.1: Position of a body described in terms of rigid body motion of the reference frame and internal deformation described using elastic coordinates [35].

The global coordinates of an arbitrary point on a rigid body can be described as [35]:

$$r_p^i = R^i + A^i \bar{u}_p^i$$

Where:

- $r_p^i$  = Global position of point P
- $R^i$  = Global position of reference coordinate
- $\bar{u}_p^i$  = Local position vector of point P
- $A^i$  = Transformation matrix

In case of a multibody system like a vehicle suspension, the reference coordinate can undergo large, non linear rigid body motion while the elastic coordinates undergo small, linear deformations relative to the local reference frame[28]. For a flexible body, the vector  $\bar{u}_p^i$  can be written as:

$$\bar{u}_p^i = \bar{u}_o^i + \bar{u}_f^i$$

### 2.2 Modal Flexibility

For the inclusion of component flexibility into a multibody system, a finite element model is unsuitable as the degrees of freedom are orders of magnitude higher than a multibody system. Modal reduction is based on the principle of capturing deformation behavior of a finite element model using a much smaller number of modal degrees of freedom. One of the definitive papers in this field is [9] which details the *Craig-Bampton method* of

modal reduction. This involves the decomposition of a component into *constraint modes* and *normal modes*. The constraint modes are obtained by producing a unit displacement of each boundary degree of freedom in turn with all other boundary degrees of freedom fixed. The normal modes of the component are used to define the motion of interior degrees of freedom relative to the fixed boundaries as normal modes of free vibration.

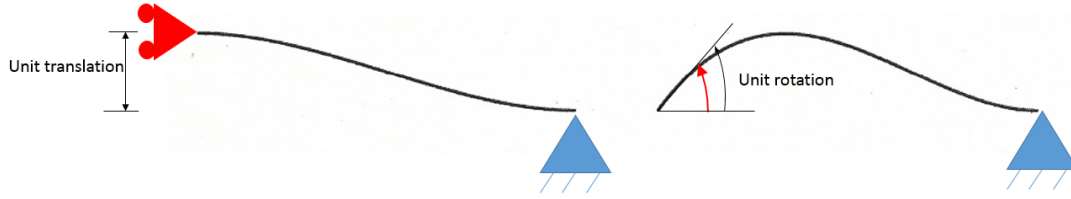


Figure 2.2: *Two constraint modes of a cantilever beam corresponding to unit translation and unit rotation of the free end (Adapted from [28]).*

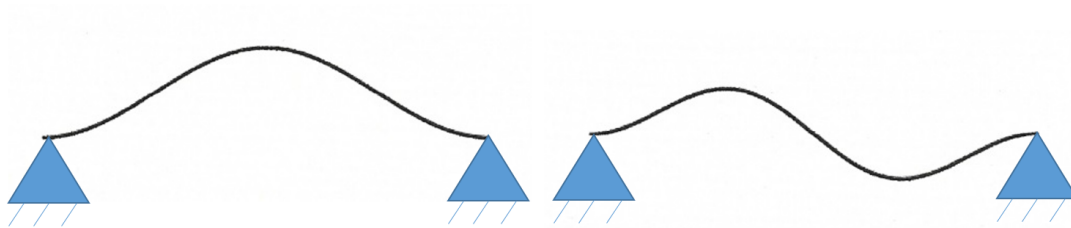


Figure 2.3: *Two normal modes of a cantilever beam (Adapted from [28]).*

Specifically, with a beam model in the floating frame of reference, the rigid body modes would be directional deformation of one end while the reference end is fixed as shown in Figure 2.2. The normal modes would be mode shapes from a fixed-fixed vibration analysis as shown in Figure 2.3. The rigid body modes can then be used to provide the deformation at one end of the beam with respect to the reference end while the normal modes would provide the shape of the beam in between the two nodes.

## 2.3 Implementation in Dymola

### 2.3.1 Model structure

All models in Dymola can be arranged in a hierarchical database called a *Library*. A library can then have multiple *Packages* which can contain various sub-packages, models or components. The library structure for the *Modelon FlexBeam Library*, developed as part of an earlier master thesis is shown in Figure 2.4.

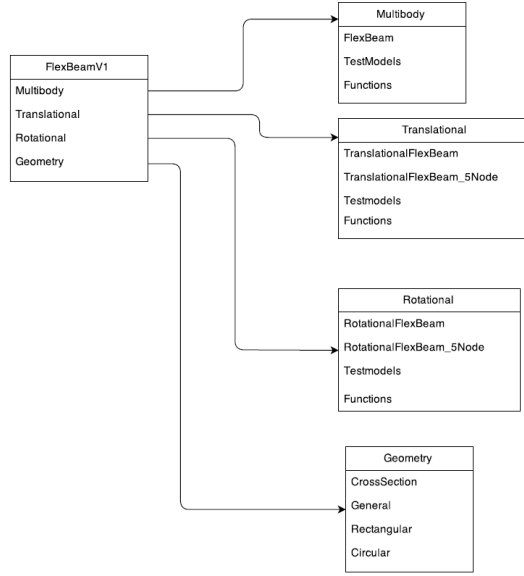


Figure 2.4: *Library structure of the Modelon FlexBeam library [17].*

The Multibody FlexBeam component was used in this Master thesis, which can deform in all 6 degrees of freedom.

### 2.3.2 FlexBeam component

The diagram layer view of the FlexBeam component is shown in Figure 2.5. It consists of two *frame* components on either end of the beam which enable exchange of position and orientation information with other Modelica multibody components and coordinate system transformations to get forces and displacements in the beam's local coordinates.

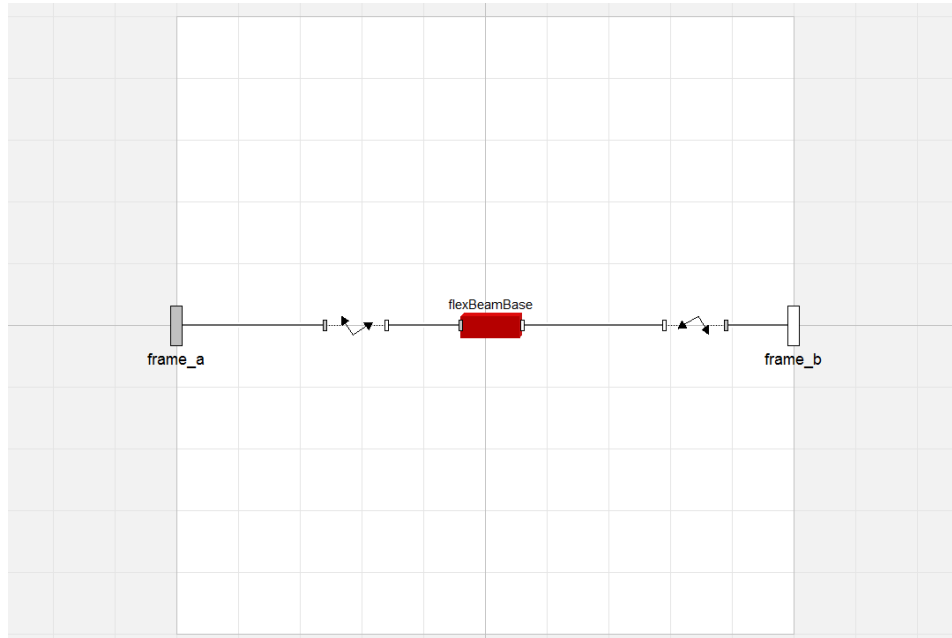


Figure 2.5: *Diagram layer view of the Multibody FlexBeam component.*

When using this component as part of a larger mechanism, an icon is visible in the diagram layer, showing the available connections as shown in Figure 2.6. This enables the drag and drop use of the FlexBeam component.

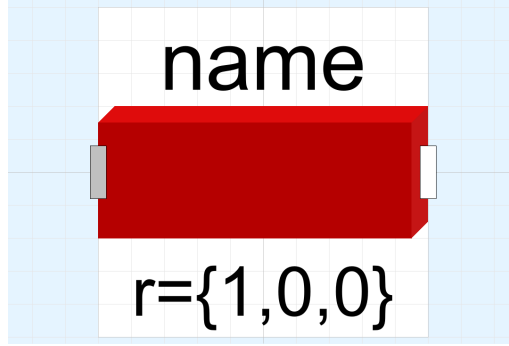


Figure 2.6: *Icon layer view of the Multibody FlexBeam component.*

### 2.3.3 FlexBeamME component

Another component from the FlexBeam library is FlexBeamME which is simply a discretized FlexBeam component. The user can define the number of elements to be used and the internal connections are handled automatically. This component was used to get equivalent discretization in both the ABAQUS beam and the FlexBeam in Dymola.

### 2.3.4 FlexBeamBushings component

During the initial evaluations, a FlexBeamBushings component was also used which is a FlexBeam component with two compliant bushing components attached to either end. These bushings provide a finite stiffness at the FlexBeam connection as is present in a welded or a bolted connection. The diagram layer view of this component is shown in Figure 2.7.

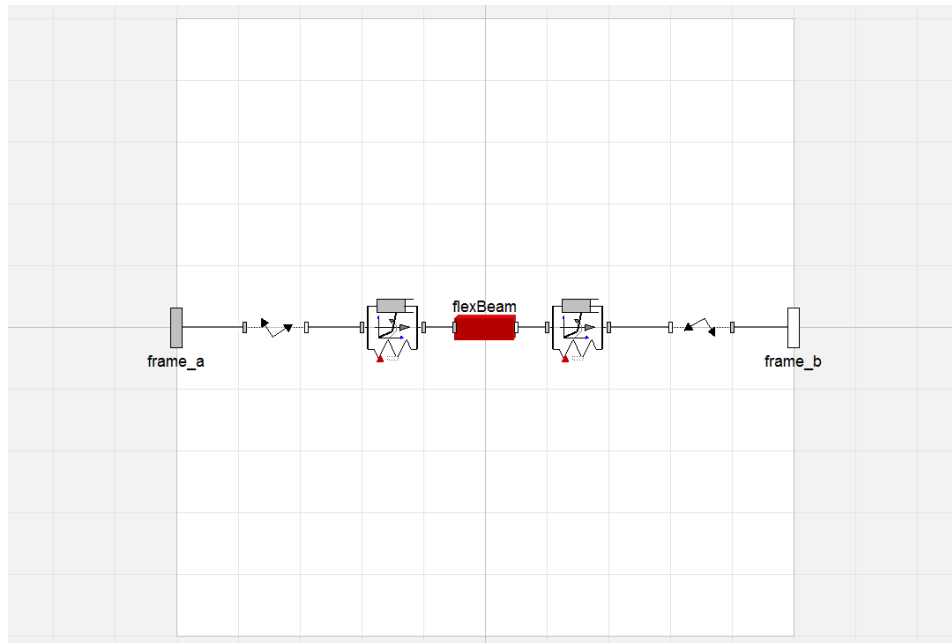


Figure 2.7: *Diagram layer view of the Multibody FlexBeamBushings component.*

## 2.4 State selection in VDL

Since Dymola in general and the Modelica multibody library in particular relies on index reduction of differential algebraic equations, careful modeling of kinematic loops like vehicle suspension is required which would enable computationally efficient solution. Most general purpose multibody codes use pure numerical approximation

(e.g. Newton-Rhapson method) and do not use DAE solvers. This also requires the number of states in a mechanical system to be equal to the degrees of freedom. To leverage this capability of Dymola, advanced joints have been designed in VDL which allow a user to manually activate and deactivate states in the different joints in a vehicle suspension. An example from a double wishbone type suspension is shown in Figure 2.8 [5]. The red arrows are used to denote states and the blue arrows are used to denote degrees of freedom. For the kinematic suspension, if the orientation and angular velocity of either the upper or lower wishbone inboard joint is known, the vertical degree of freedom (position and velocity) at the wheel can be specified (The orientation about the vertical axis is defined using the steering linkage position). For the elasto-kinematic suspension, ten additional degrees of freedom are added in the form of compliances, requiring 22 states to be selected out of 24 available states. This is achieved by selecting 11 positions and their derivatives as states.

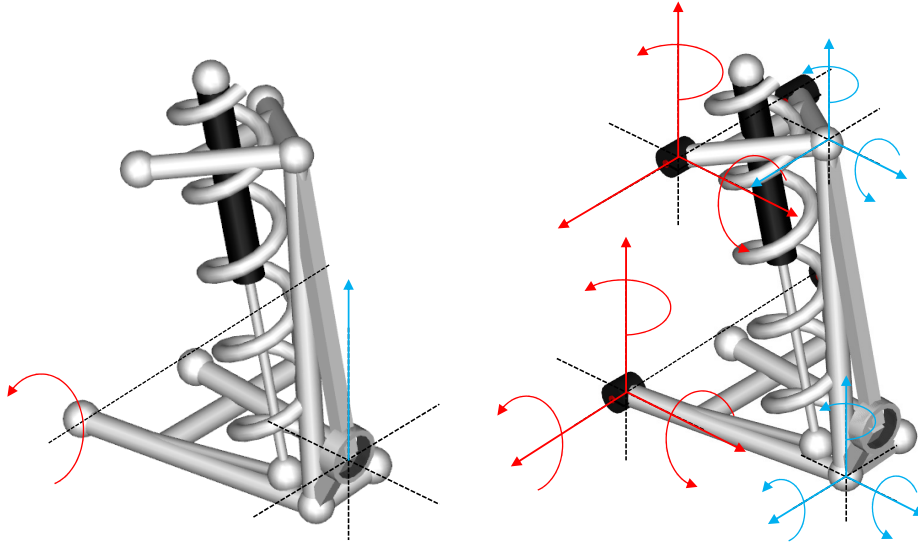


Figure 2.8: *State selection and degrees of freedom for a kinematic and elasto-kinematic double wishbone suspension (Adapted from [5]).*

The above discussion is for state selection in a mechanism being user defined in Dymola. This is done by setting the flag `StateSelect.always` to true in a joint or a body. It is also possible to let Dymola decide which states to use in a given mechanism during the course of a simulation i.e. dynamic state selection. In such a case, the status of the `StateSelect` flag defines the preference to be used by the Dymola compiler (`StateSelect.prefer` to be used before `StateSelect.default` which is in turn preferred over `StateSelect.avoid`). There is also the option to disable state selection altogether (`StateSelect.never`) in a component. For best computational performance, dynamic state selection should be avoided in Dymola.

## 2.5 Twist Beam Suspension

A twist beam suspension is a semi dependent suspension consisting of two trailing arms connected with a cross beam, usually welded to the arms as shown in Figure 2.9. The roll stiffness of the axle is provided by large elastic deformations of the cross beam. Due to component integration into one major assembly, weight and cost benefits, this type of suspension is widely used in the rear of small cars over the world despite its limitations in kinematic and compliance performance [20]. The calculation of the basic kinematic and compliant characteristics is shown in [34].

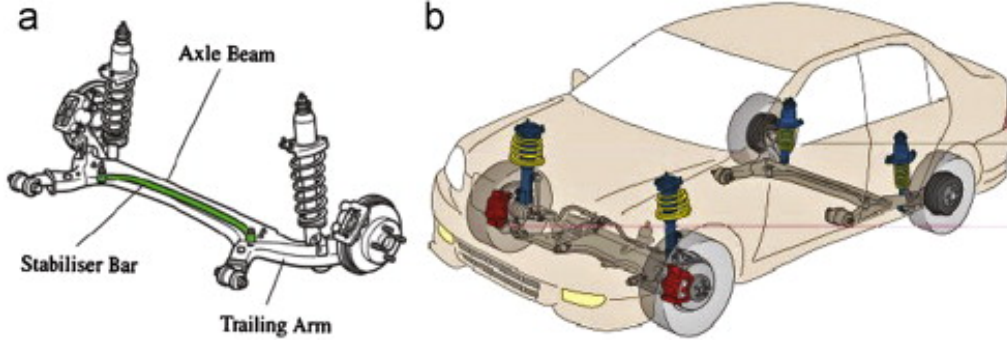


Figure 2.9: Descriptions of twist beam axles (a) twist beam rear suspension (b) suspension system for a typical front wheel drive vehicle [21].

### 2.5.1 Current state of the art

Due to the inherent coupling of the different degrees of freedom in this type of axle, the design process used by OEMs and Tier I suppliers relies on a combination of multibody simulations using flexible bodies and finite element analysis. The paper [18] has a comparison of a multibody model of a twist beam using modal reduction and a finite element (FE) model. It can be seen that a multibody model is valid for most of ride and handling evaluation and is computationally much faster than a FE model. Recent advances enable the use of non linear flexible bodies which should provide increased accuracy for large deformations [29]. However, the geometry data to make such a FE model is not available until the detail design stage and the modal reduction process from the FE data is still rather time consuming.

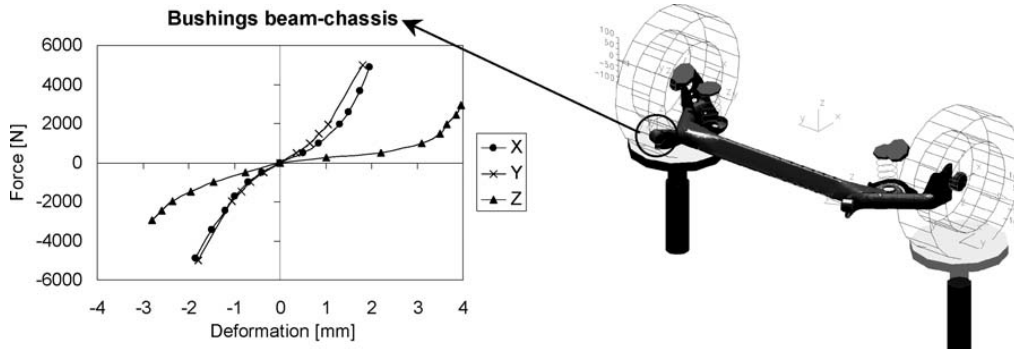


Figure 2.10: A multibody model of a twist beam suspension made in SIMPACK [18].

### 2.5.2 Implementation in VDL

Further abstraction of the twist beam suspension can be used for concept studies, hardpoint and bushing stiffness/orientation definition as well as parameter sensitivity analysis of vehicle dynamics performance metrics. VDL is particularly suitable environment for such studies as the scaling of model detail is very flexible based on the template based Modelica models. Also, the realtime capabilities [13, 16, 36] and the ability to leverage on the FMI technology [19] make a detailed and computationally efficient vehicle dynamics model available earlier in the design cycle across engineering domains e.g control engineering and powertrain engineering.

The major contributions of the different design parameters on suspension characteristics for a specific case can be seen in [8]. In this case the compliance from the bushing, the bearing and the spring have major effect on the lateral force compliance steer, compliance camber, lateral compliance, longitudinal steer and longitudinal compliance. This goes on to show that significant improvement in the model fidelity can be obtained by using chassis mount bushings and compliant hubs. These characteristics can be easily modeled in VDL by using available components. The use of a flexible cross beam and trailing arms provides further accuracy in roll stiffness and roll steer coefficient as well as lateral and longitudinal compliance.

For the purpose of this thesis, a twist beam was modeled in VDL using a flexible cross beam and validated against a similar model in ADAMS/Car which is a multibody modeling and simulation environment by MSC Software, tailored for road vehicle applications [32].

The choice of a cross section geometry for the cross beam is important as it enables the beam to provide toe and camber stiffness while still being relatively soft in roll. Typical cross sections are open U or V type cross section [22, 34] and more recently semi-open profiles are being used to get higher specific roll stiffness values [21]. For the purpose of this thesis, the geometric properties of an open V type section have been used [22, 23] to get realistic bending and torsional stiffness of the cross beam. An advantage of using a flexible cross beam is also in the parameter sensitivity analysis of cross beam location as well as the orientation as shown in [21, 37]. With the available parametrization in VDL such studies can be carried out with good computational efficiency.

## 2.6 Output variables of suspension tests

A multibody model of a vehicle suspension provides an insight into the geometric position and orientation of the wheel with wheel travel as well as the directional loading of the contact patch [7]. A kinematics and compliance test rig [6] is exclusively used to perform tests on an actual vehicle by using kinematic or force inputs at the wheel center or the contact patch. Most MBS software including ADAMS/Car and VDL have test rigs replicating such quasi-static tests on suspension assemblies or full vehicles.

Some of the most common output variables were compared as part of this thesis [7]. Though this list isn't exhaustive, it gives a good indication of the suitability of the beam model used to evaluate vehicle suspension behavior. The following variables were evaluated with respect to wheel travel:

- Wheel center longitudinal movement.
- Track change.
- Toe angle.
- Camber angle.
- Actuator reaction force, which provides an indication of wheel rate.
- Reaction forces on chassis mount, to ensure the forces and displacements from the wheel were correctly transferred to the chassis.

The following variables were evaluated with respect to force inputs at the wheel center or contact patch [6]:

- Aligning torque steer and camber compliance.
- Lateral Force - deflection, steer, and camber compliance.
- Longitudinal Force - deflection, steer, and camber compliance.
- Reaction forces on chassis mount, to ensure the forces and displacements from the wheel were correctly transferred to the chassis.

The definitions of these variables depend on the coordinate system being used and most of the parameters have been measured in this thesis as shown in [30], with differences explained wherever applicable.

In addition to these tests a multibody model is also useful for understanding component loading environment as a function of road loads. The loads on the suspension components as well as the body mountings can be used as input to the respective finite element models to evaluate component stiffness, fatigue life as well as ultimate strength based on so called "abuse loadcase" [7]. This aspect can be evaluated by measuring the reaction forces on the different attachment points for static loads at the contact patch. A step further is to run a full vehicle model on a virtual proving ground and use the time domain forces for evaluating fatigue life.

Finally, the dynamic performance of a vehicle suspension in the frequency domain can be measured by applying a frequency sweep at the contact patch. The measured displacement or acceleration of the wheel center and the body provide a measure of ride comfort and vertical wheel load variation [7]. The former is a critical part of the NVH (Noise, Vibration and Harshness) perception, while the latter provides an estimate of change in available grip levels depending on road inputs which is interesting for performance envelope and active safety evaluations.

## 3 Method

### 3.1 Twist Beam Modeling in ADAMS/Car

#### 3.1.1 Coordinate system

The coordinate system used in ADAMS Car is Backward-Rightward-Upward with the origin located at the vehicle center line in front of the front axle as shown in Figure 3.1.

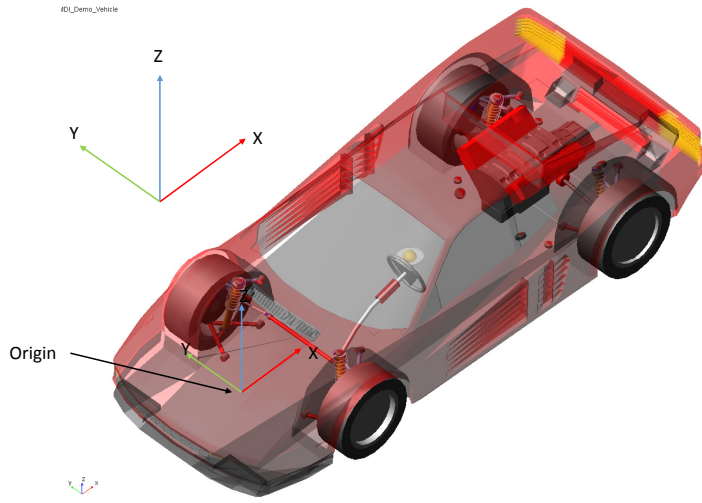


Figure 3.1: *Coordinate system in ADAMS/Car (MDI\_Demo\_Vehicle shown).*

#### 3.1.2 Template modification

The ADAMS/Car package relies on suspension templates which define the topology and default hardpoints, bushing stiffness, spring and damping properties. This template is then used as a basis to model a subsystem which has parametrized properties which can be modified and further used as part of a suspension assembly. One of the available templates in the default installation of ADAMS/Car (2014.0.1) is the `_twist_beam.tpl` template as shown in Figure 3.2.

The standard file format used for importing flexible bodies modeled using an external FE code is MNF (Modal Neutral File) in ADAMS. Since this model uses a single ADAMS MNF flexible body for the trailing arms and the cross beam, it was modified to have rigid trailing arms connected with a beam element flexible body so that a model with known cross section parameters for the cross beam could be modeled in VDL as well. This also enabled the use of an ABAQUS Euler-Bernoulli beam to generate the MNF for the ADAMS model. The hardpoint for the cross beam location was chosen arbitrarily to be at 40% of the bushing- wheel center distance to enable both bending and torsion effects in the cross beam. The MNF component was connected to the trailing arm using an interface part and a fixed joint. The modified template is shown in Figure 3.3.

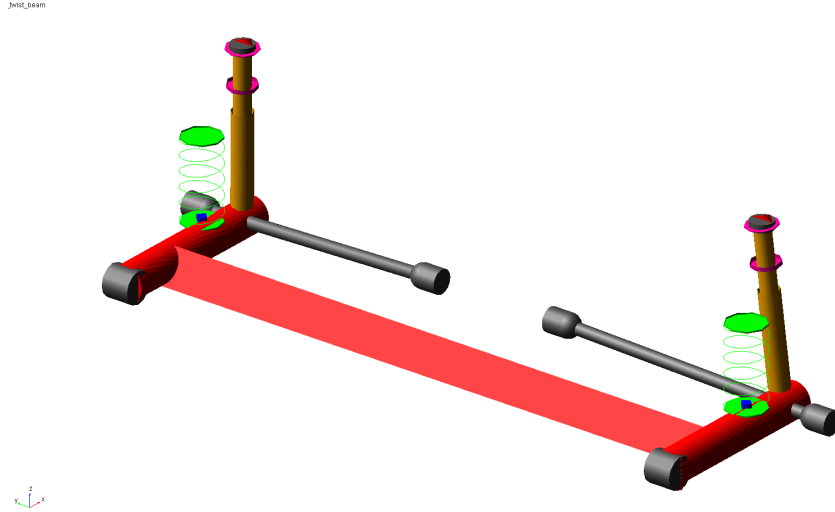


Figure 3.2: *\_twist\_beam.tpl ADAMS/Car template.*

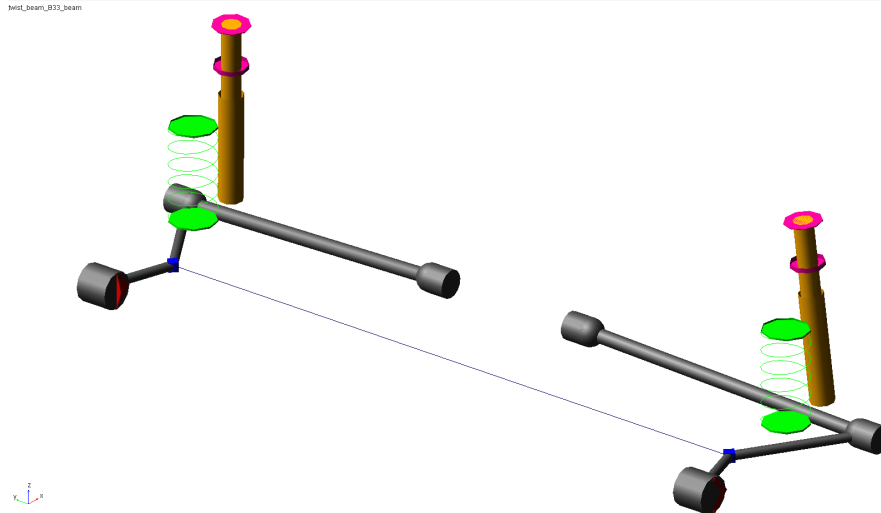


Figure 3.3: *Modified ADAMS/Car template.*

To generate the MNF file for the ADAMS model, ABAQUS 2016 was used to model the beam using Euler Bernoulli formulation and then modal reduction and flexible body generation was performed using ABAQUS inbuilt functions as described in Section 3.2.4.

### 3.1.3 Hardpoints

The hardpoints in ADAMS/Car contain location information only and define the basic layout of the components, based on which the topology is defined. The naming convention followed for all features in ADAMS/Car is **feature type** followed by **side** and then **feature name**. For example, **hp1\_spring\_upper** is a left hardpoint for the spring upper attachment points. The various hardpoints from the template are shown in Figure 3.4 and the location data for the left side is shown in Table 3.1. The right side hardpoints are mirrored about the X axis.

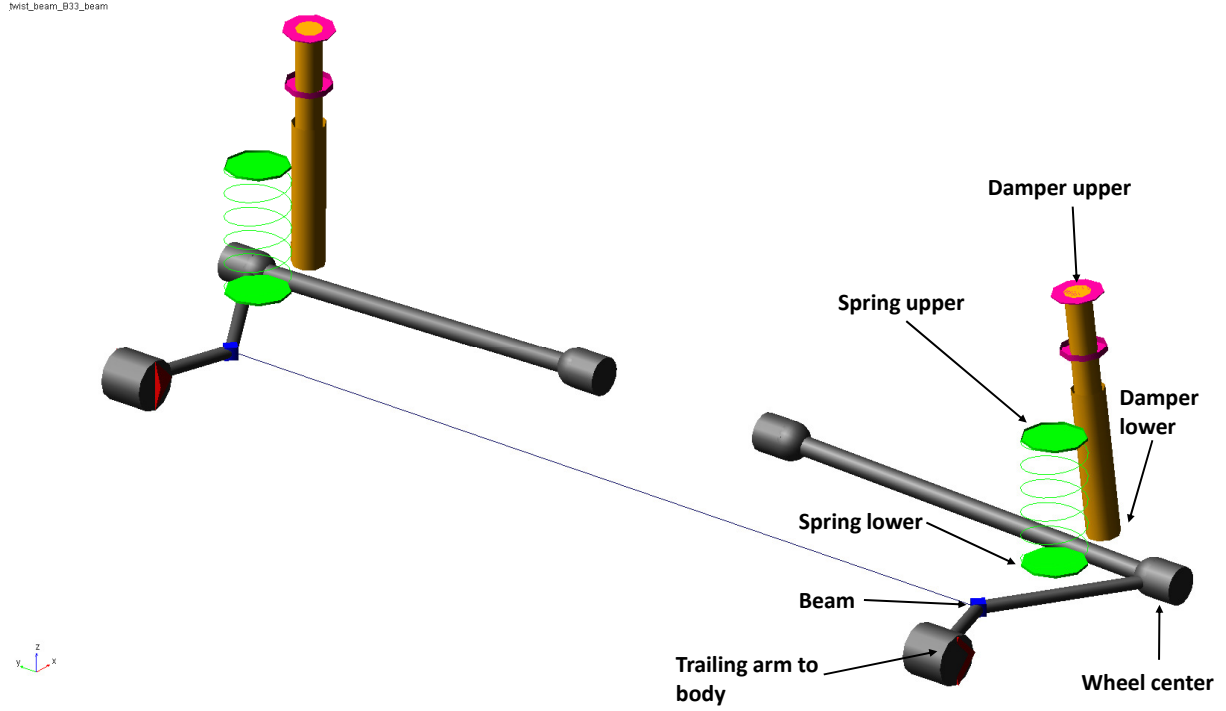


Figure 3.4: *Description of hardpoints used in the ADAMS/Car model.*

| Hardpoint             | X(mm)   | Y(mm)   | Z(mm)  |
|-----------------------|---------|---------|--------|
| hpl_beam              | -240    | -700    | 500    |
| hpl_damper_lower      | 1.65    | -744.11 | 542.09 |
| hpl_damper_upper      | -28.34  | -719.11 | 942.09 |
| hpl_spring_lower      | -117.17 | -745.61 | 542.27 |
| hpl_spring_upper      | -117.17 | -745.61 | 742.27 |
| hpl_trailing_arm_body | -400    | -750    | 500    |
| hpl_wheel_center      | 0       | -850    | 500    |

Table 3.1: Hardpoints used in the twist beam template (left side).

### 3.1.4 Spring

The spring used in the model is one of the available springs in ADAMS/Car. A more realistic value for a compact car can also be used from [6] to match the wheel rate but a default model was used from the ADAMS database to compare with the VDL model for the purpose of this thesis. It is a non linear spring with the non linearity caused by the definition of a bump stop using the spring stiffness. This prevents early "bottoming out" of the suspension when used in a vehicle. The force deflection characteristics are shown in Figure 3.5. The spring preload is defined using the installed length as  $130mm$ .

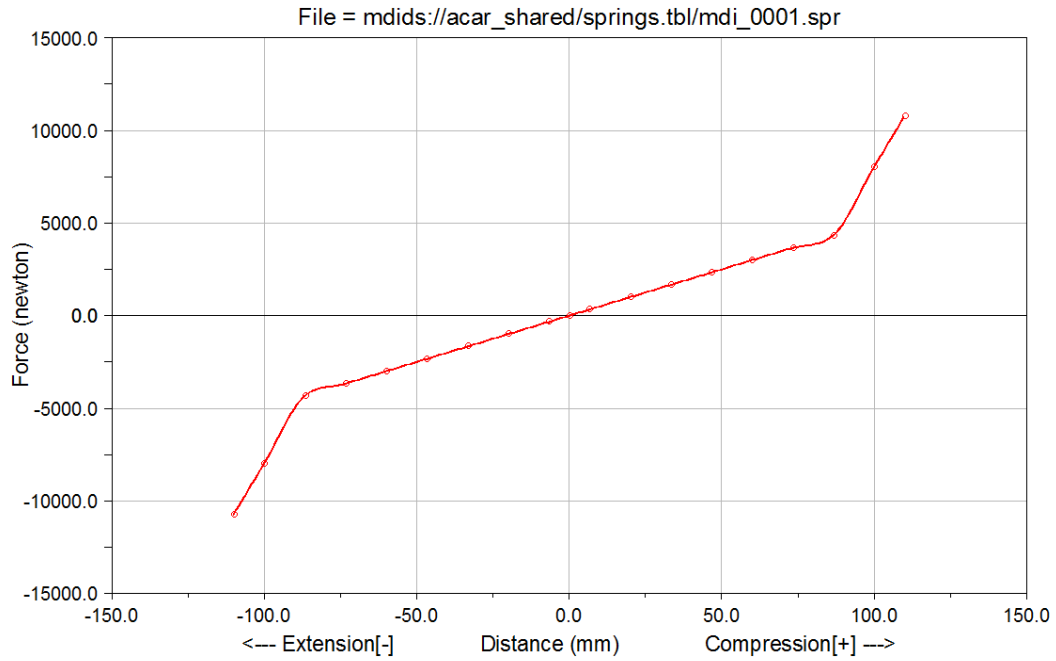


Figure 3.5: *Force deflection characteristics of the spring used in the model.*

### 3.1.5 Damping

The damper used in the model is one of the default ADAMS/Car dampers and the force velocity characteristics are shown in Figure 3.6. An additional bump stop is also present in the ADAMS model, which is coaxial with the damper. It has a clearance of  $90\text{mm}$  from the top mount as shown in Figure 3.7.

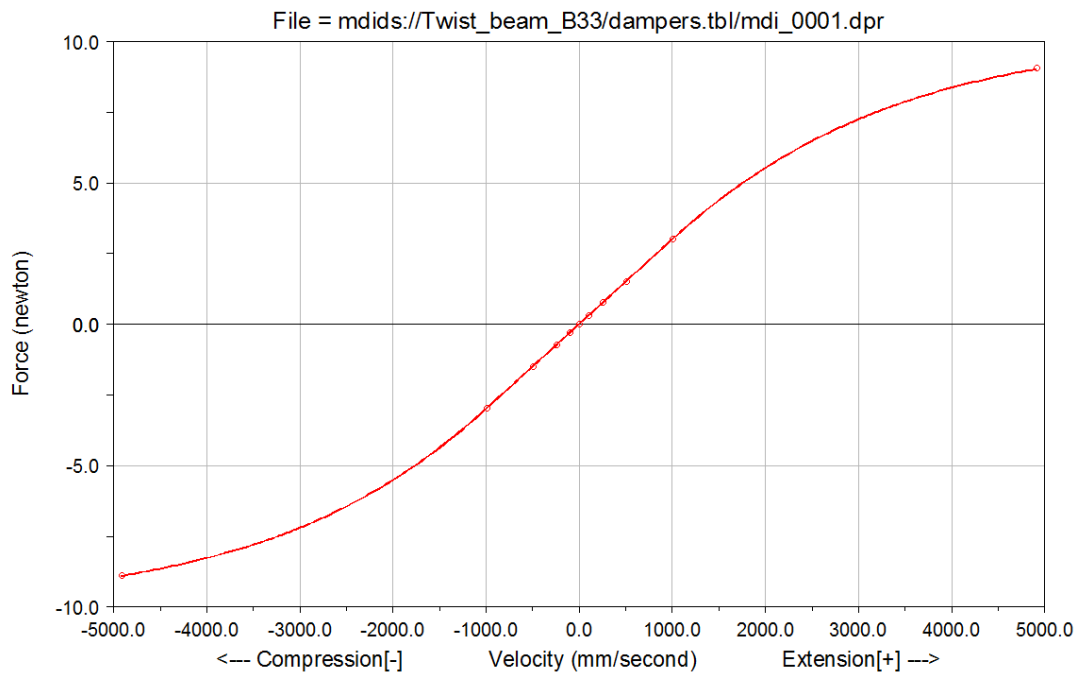


Figure 3.6: *Force velocity characteristics of the damper used in the model.*

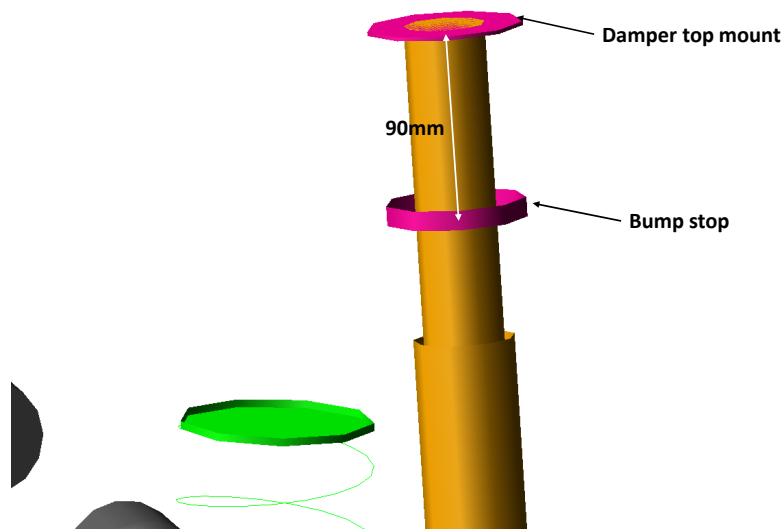


Figure 3.7: *Bump stop definition with the damper.*

### 3.1.6 Bushing properties

The bushing used for the chassis attachment point is a linear bushing relatively soft in twist around its axis and stiff in all the other degrees of freedom. The coordinate system is shown in Figure 3.8 in orange. The translation and rotational stiffness characteristics are shown in Figure 3.9 and Figure 3.10 respectively and the damping properties are shown in Table 3.2.

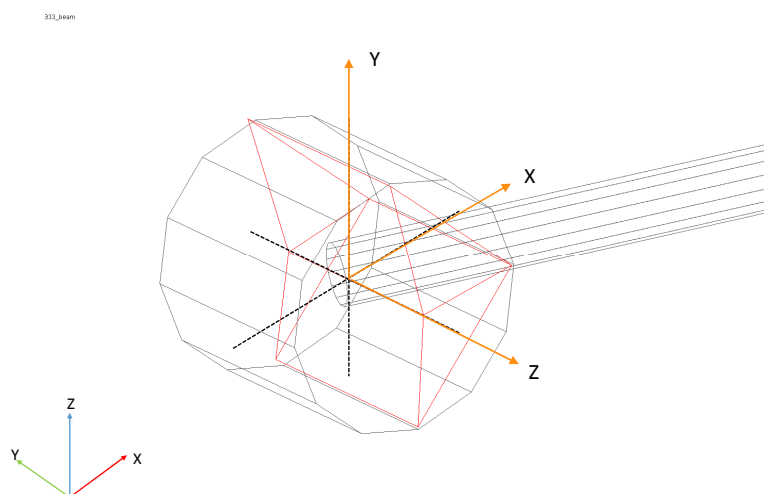


Figure 3.8: *Coordinate system of ADAMS/Car bushing.*

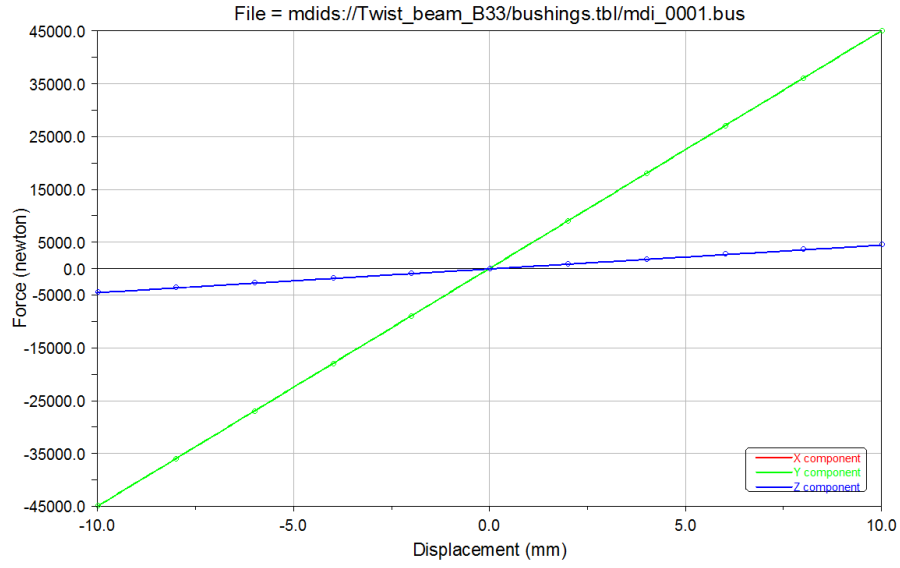


Figure 3.9: *Translation stiffness of ADAMS/Car bushing.*

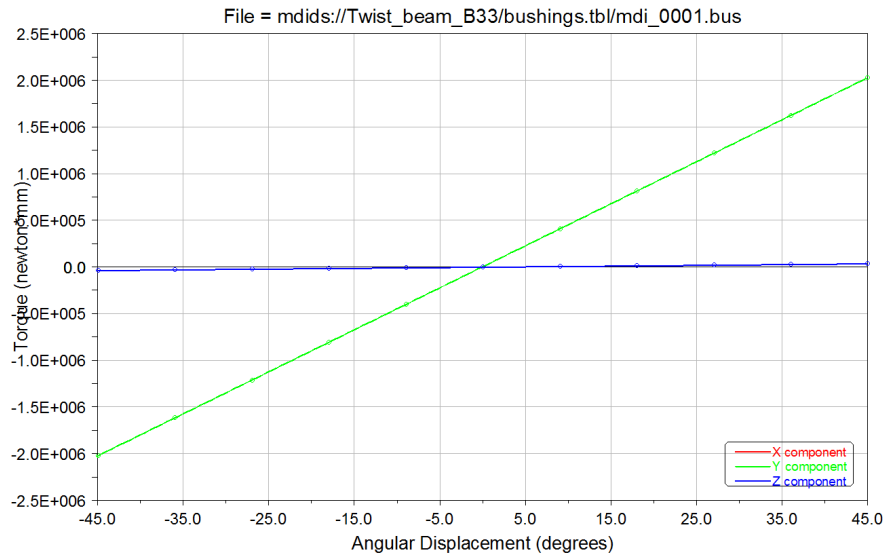


Figure 3.10: *Rotation stiffness of ADAMS/Car bushing.*

Table 3.2: Damping properties of ADAMS/Car bushing.

| Direction                           | X   | Y   | Z   |
|-------------------------------------|-----|-----|-----|
| Translational damping ( $Nsec/mm$ ) | 0.5 | 0.5 | 0.5 |
| Rotational damping ( $Nmmsec/deg$ ) | 0.5 | 0.5 | 0.5 |

### 3.1.7 Suspension assembly and test rig

The default suspension test rig MDI.SDI.TESTRIG was used for all suspension experiments. The suspension assembly with the test rig is shown in Figure 3.11

twist\_beam

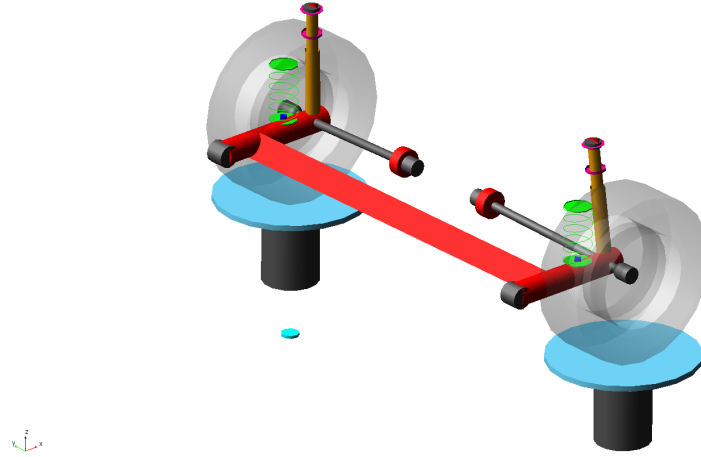


Figure 3.11: *ADAMS/Car suspension and test rig assembly.*

### 3.1.8 Tire properties

The wheel and tire model used in the test rig is a rigid wheel and the relevant properties are the tire stiffness and radius as shown in Table 3.3. The corresponding ADAMS/Car dialog is shown in Figure 3.12. The rest of the properties are used for dynamic analysis and calculation of "anti" geometry properties, roll stiffness and roll rate [30] and were left at the default values as they are not directly compared to VDL results. The radius is calculated corresponding to a 225/40 R16 tire in VDL.

Table 3.3: Tire properties used in the ADAMS/Car test rig.

| Parameter                     | Value  |
|-------------------------------|--------|
| Tire stiffness ( $N/mm$ )     | $10^6$ |
| Tire unloaded radius ( $mm$ ) | 293.2  |

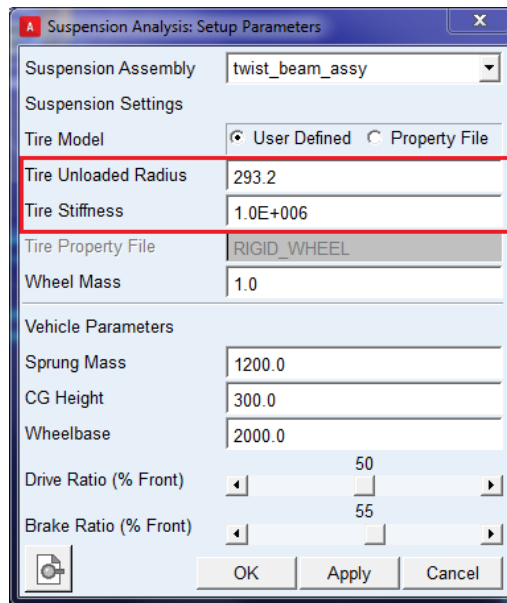


Figure 3.12: *Dimensions and stiffness properties of ADAMS/Car tire used.*

## 3.2 Beam Modeling in ABAQUS

Though ADAMS native flexible components are available in the form of FE/Part and ViewFlex [28], it was decided to use a MNF file, generated using ABAQUS as an Euler-Bernoulli formulation, valid for linear deformations is the closest in implementation to the Dymola beam. The preprocessing environment used was ABAQUS/CAE 2016.

### 3.2.1 Beam cross section

Since the cross beam is needed to be relatively soft in torsion while still being stiff in the bending directions, typically V or C type cross sections are used [34]. More recently semi-open type profiles are being used to get increased roll stiffness while still maintaining weight targets for the assembly [21]. Exact dimensions of the entire torsion beam assembly as well as cross sections are proprietary to OEMs and tier 1 suppliers but some studies are available, one of which [22, 23] has a comparison of effect of cross section properties on vehicle suspension characteristics. One of the cross sections from this study is used as shown in Table 3.4.

Table 3.4: Cross section properties.

| Parameter                             | Value         |
|---------------------------------------|---------------|
| Area ( $mm^2$ )                       | 927           |
| Moment of Inertia about x $I_x(mm^4)$ | $6.65 * 10^5$ |
| Moment of Inertia about z $I_z(mm^4)$ | $6.87 * 10^5$ |
| Torsional constant $K(mm^4)$          | 6212          |

To model this cross section, a generalized cross section was used in ABAQUS/CAE and the relevant dialog box is shown in Figure 3.13. The sectorial moment  $\Gamma_O$  and the warping constant  $\Gamma_W$  were left to be 0 and the cross section offset and shear center effects were not accounted for as the Dymola beam in its current implementation does not have these effects. Despite these limitations, the beam provides realistic bending and torsion stiffness required for its use in a twist beam suspension assembly.

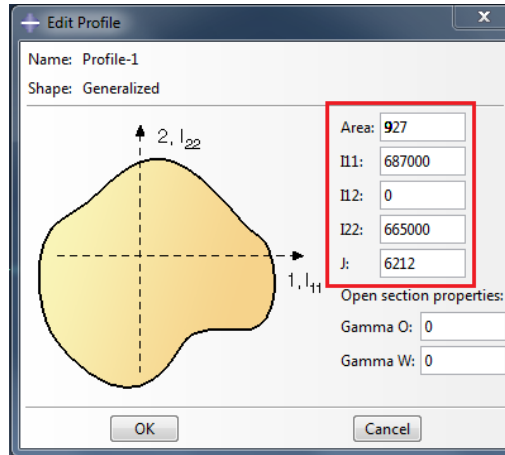


Figure 3.13: Generalized cross section definition in ABAQUS/CAE.

### 3.2.2 Material properties

The material used for modeling the cross beam is steel and the relevant properties are shown in Table 3.5 and the ABAQUS/CAE dialog box is shown in Figure 3.14.

Table 3.5: Material properties used.

| Property                     | Value            |
|------------------------------|------------------|
| Density ( $kg/mm^3$ )        | $8.05 * 10^{-6}$ |
| Young's Modulus ( $N/mm^2$ ) | $2.1 * 10^5$     |
| Poisson's ratio              | 0.3              |

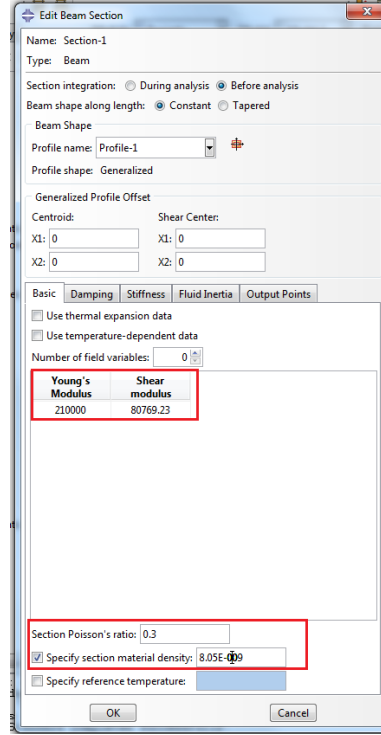


Figure 3.14: Beam material property definition in ABAQUS/CAE.

### 3.2.3 Beam element type and orientation

As described in the ABAQUS analysis manual [10], the element type used for modeling an Euler Bernoulli beam having 6 degrees of freedom is B33. 10 elements were used to mesh the 1.4m long beam.

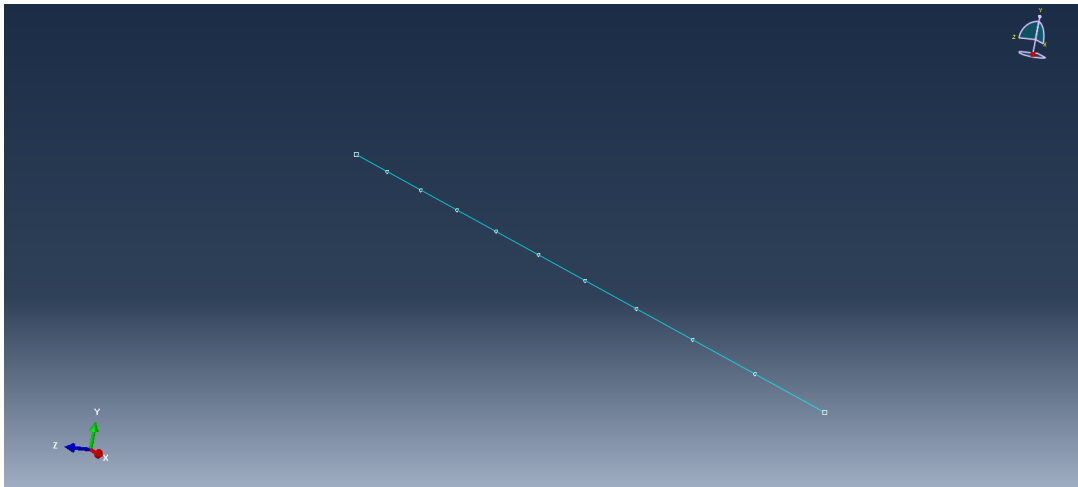


Figure 3.15: Beam node and element visualization in ABAQUS/CAE.

Since the coordinate system used in [22, 23] is the ADAMS/Car coordinate system, the beam cross section orientation was defined to get the correct area moment of inertia about the  $I_{22}$  and  $I_{11}$  axis corresponding to X and Z axis as shown in Figure 3.16.

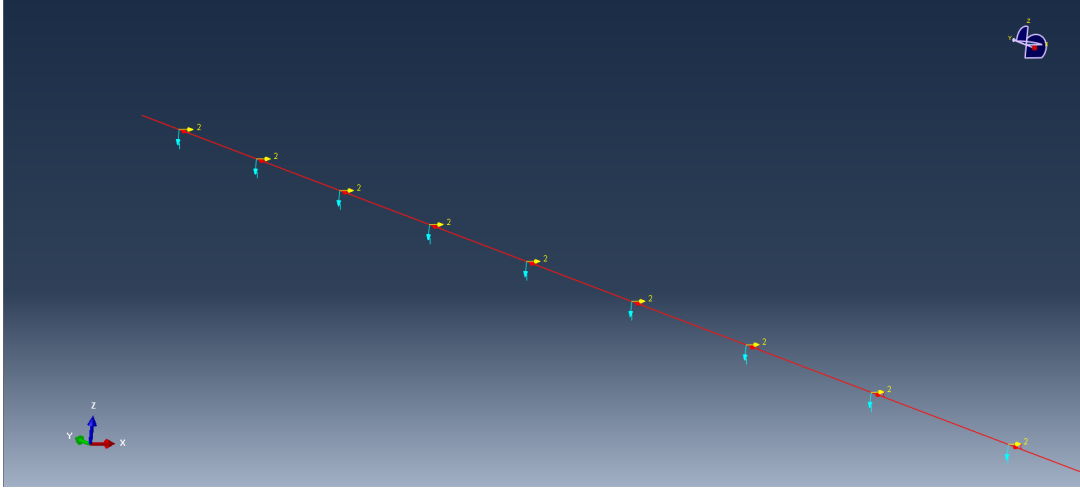


Figure 3.16: *Beam cross section orientation in ABAQUS/CAE.*

### 3.2.4 Modal reduction in ABAQUS/CAE

Since the MNF format in ADAMS uses the Craig-Bampton method, the generation of a MNF file from ABAQUS relies on three steps [10]:

- A fixed-fixed normal modes analysis.
- Substructure generation using the normal modes analysis.
- Using the ABAQUS to ADAMS translator to convert the substructure into a MNF file.

The input deck for generating such a MNF file is shown in Appendix A.1. For the normal modes analysis, the default number of 20 modes was used. When using the ABAQUS to ADAMS translator, 12 additional modes are added to the MNF, resulting in 32 MNF modes. For each interface node, 6 constraint modes are used and one of the interface nodes acts as the reference point for rigid body motion. The results from the substructure generation process can also be checked using one of the output files (\*.dat), which is shown in Appendix B.1. It has to be noted that before being used in ADAMS, a further modification of these modes is done in terms of mode shape orthonormalization [28] which explains the difference in the eigenvalues as seen by querying a flexible body's properties from within ADAMS and the FE model. By applying the fixed-fixed boundary conditions again in ADAMS and finding the eigenvalues and eigenvectors using ADAMS/Linear, the results can be compared with the original FE model.

## 3.3 Twist Beam Modeling in VDL

### 3.3.1 Coordinate system

The coordinate system used in VDL is Forward-Leftward-Upward, based on the ISO 8855 standard as shown in Figure 3.17.

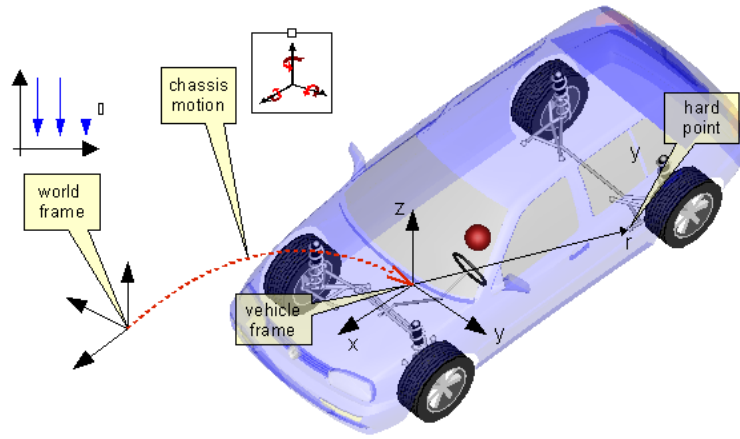


Figure 3.17: *Coordinate system in VDL [27].*

### 3.3.2 Linkage modification

The models in VDL utilize a similar template based approach for defining axles and vehicles. The basic assembly used for modeling was a linkage assembly for semi-dependent suspension as shown in Figure 1.1. The changes made to the assembly, in order to compare it to the ADAMS model of comparable complexity were the following:

- Spherical joints were replaced with bushings at the chassis mount.
- The cross beam compliance component was replaced with the flexible beam.
- All the hardpoints were taken from the ADAMS model.

The modified linkage model is shown in Figure 3.18 with the changes encircled in Red.

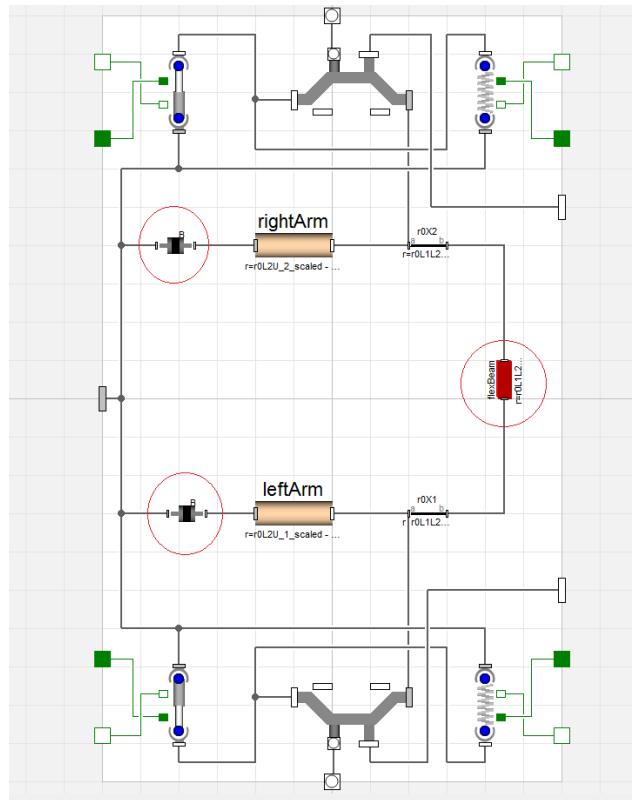


Figure 3.18: *Linkage modeling in VDL.*

### 3.3.3 Beam model

For modeling the cross beam the multibody FlexBeam component was used with all the static modes turned on and the stiffness damping set to  $1 * 10^{-3}$  in order to minimize initial transients. A general cross section was used and the section properties entered were the same as used in modeling the ABAQUS beam as shown in Figure 3.19.

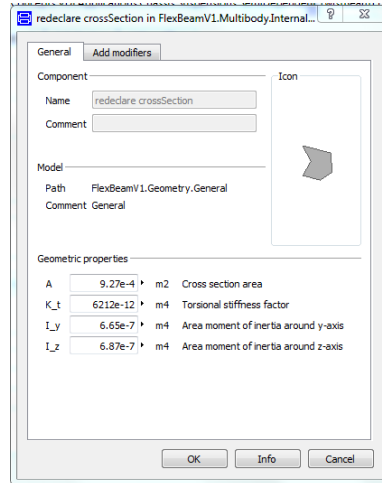


Figure 3.19: *Beam cross section properties in VDL.*

### 3.3.4 Bushings in VDL

The bushing coordinate system in VDL is shown in Figure 3.20. Additionally, in VDL version 2.3, the radial direction of the bushing can be specified which enables the use of the ADAMS/Car bushing coordinate system.

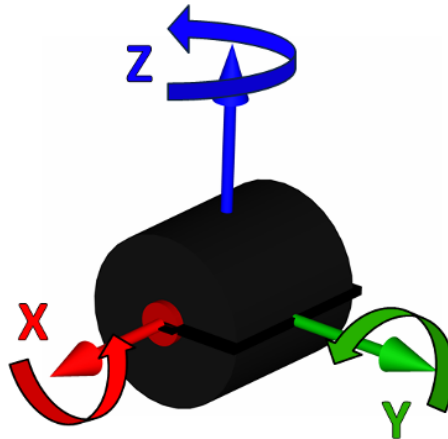


Figure 3.20: *Bushing coordinate system in VDL [27].*

### 3.3.5 Suspension Assembly

The linkage model described in the previous section was used in a suspension assembly for semi dependent suspensions. Initially, the TwistBeam suspension from VDL was modified to use the new linkage model using the linear springs and dampers to evaluate state selection and simulation performance. For comparison to the ADAMS/Car model, the respective spring and damping property files from ADAMS/Car were used instead. The suspension model is shown in Figure 3.21.

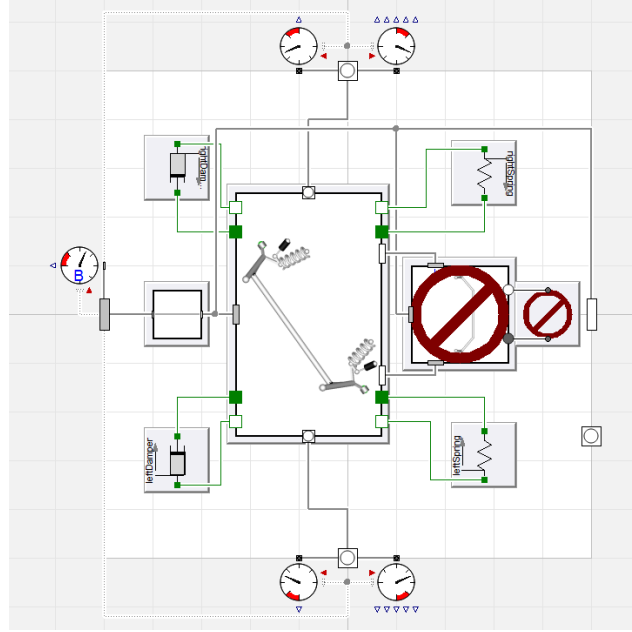
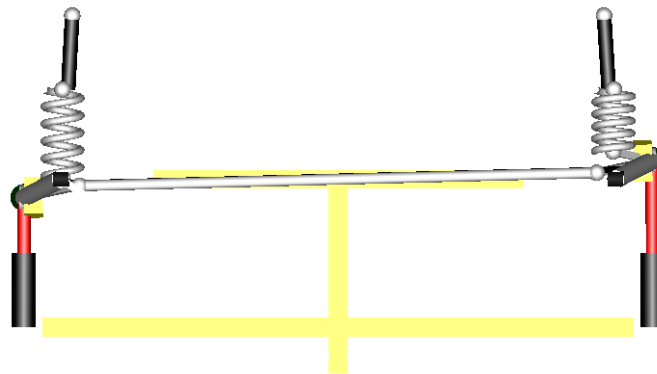


Figure 3.21: *Suspension assembly modeling in VDL.*

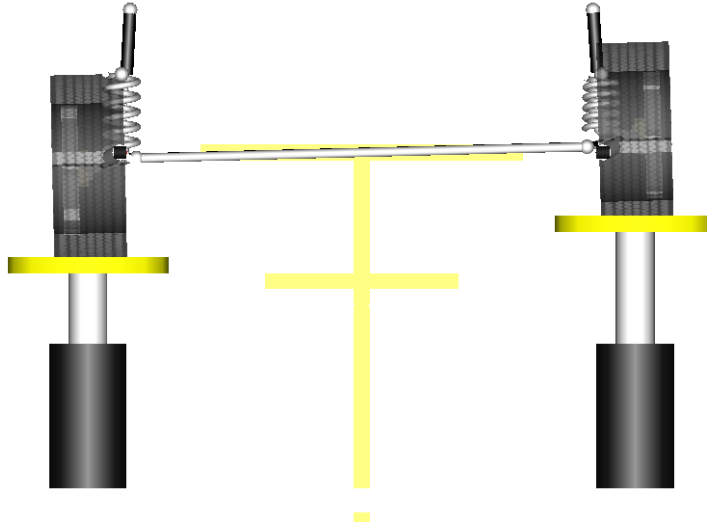
### 3.3.6 Test rigs

Two types of test rigs were used for testing the VDL model:

- Hub rig as shown in fig 3.22a.
- Wheel rig as shown in fig 3.22b.



(a) *Hub rig with twist beam suspension assembly in VDL.*



(b) *Wheel rig with twist beam suspension assembly in VDL.*

Figure 3.22: *VDL suspension test rigs.*

The difference between the two rigs is the interface of the suspension assembly with the actuator. In case of the wheel rig, the actuators act on the wheels attached to the hubs while in the hub rig, force is applied directly to the hubs. The hub rig was used for comparing the models while evaluating different state selection options and possible performance issues while using the FlexBeam component. The wheel rig was used for comparison with the ADAMS test rig results. The wheel and tire model used in the wheel rig is a rigid wheel with the tire dimension corresponding to 225/40 R16 and stiffness property same as the ADAMS tire as shown in Figure 3.12.

### 3.3.7 Importing ADAMS/Car data into VDL

ADAMS/Car data is organized either as an XML file or a TeimOrbit [30] file which is a text based format. All the ADAMS data used in this thesis is in the TeimOrbit format. The TeimOrbit files for the subsystem and assembly are shown in Appendix C.2 and C.1. VDL has a Migration package for ADAMS/Car which enabled the use of the spring, damper and bushing property from the respective TeimOrbit files. Additionally, the Modelon library in Dymola has a data access package for TeimOrbit files which was used to read the hardpoints from the subsystem file. The use of the TeimOrbit files enables faster and easier data reuse and building of "data aware" VDL models which utilize the latest data even from a different modeling environment [4]. The linkage and the suspension assembly, utilizing these components are shown in Figure 3.23.

Difference in coordinate systems of the input data and VDL can be handled by simply dragging in a vehicle frame orientation component onto the diagram layer of the experiment. Figure 3.24 shows the diagram layer view and the coordinate system selection dialog box for such an experiment.

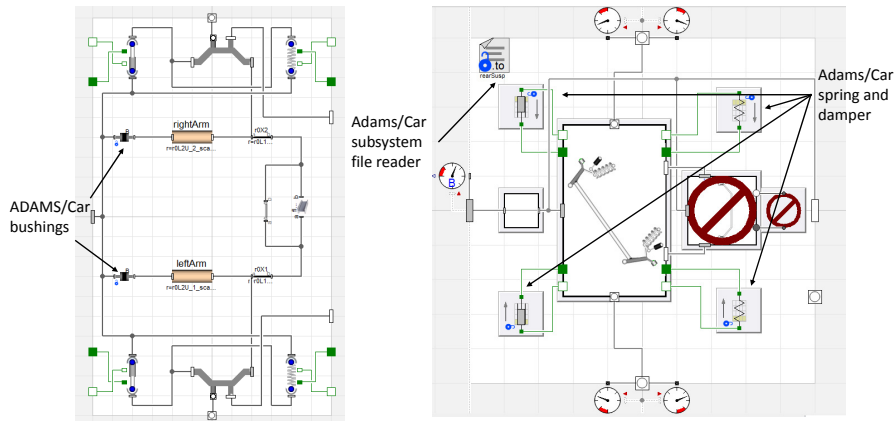
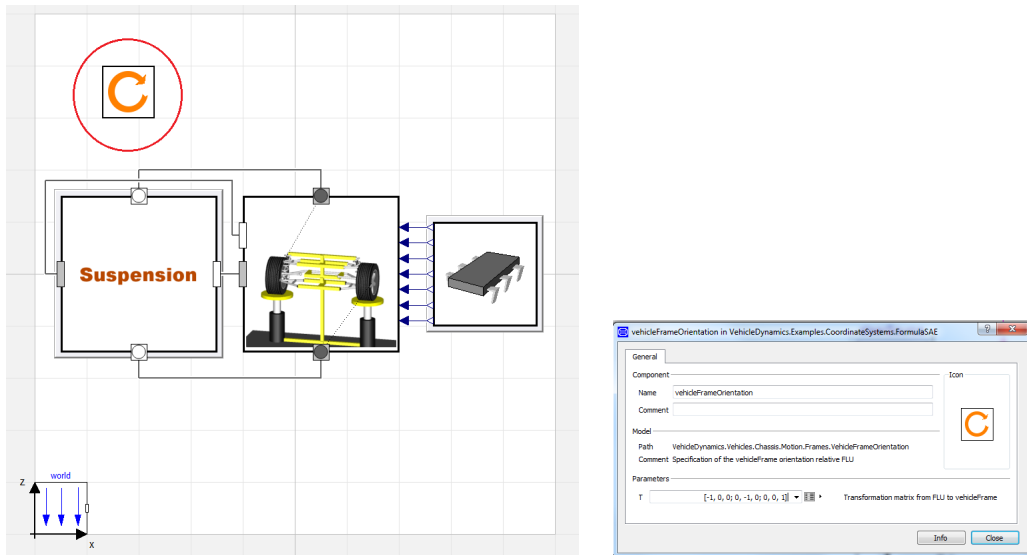


Figure 3.23: *Linkage and suspension assembly in VDL utilizing ADAMS/Car data.*



(a) *Diagram layer view of a suspension experiment with a coordinate system transform (encircled in red).*

(b) *Dialog box for selecting transformation matrix.*

Figure 3.24: *A suspension experiment in VDL utilizing input data in ADAMS/Car coordinate system.*

### 3.3.8 State selection and performance improvement

As shown in Figure 1.1, an existing twist beam model is available in VDL utilizing spherical joints at the chassis mounts and a compliance element connecting the two trailing arms. Figure 3.25 shows the degrees of freedom in blue and the states in red. Since the wheels have three degrees of freedom each, all orientations and their derivatives of the spherical joints are used as states leading to 12 states in total.

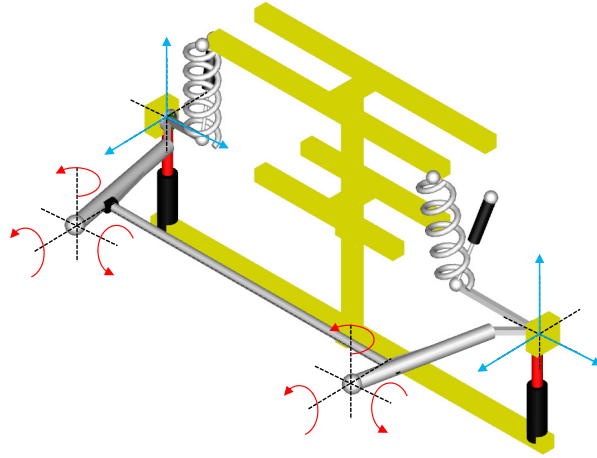


Figure 3.25: *State selection and degrees of freedom in `TwistBeamTT` VDL linkage model.*

On replacing the compliance element with `FlexBeam` component, it is desirable to use the 6 generalized deformation coordinates and their derivatives as states. On direct replacement in the `TwistBeamTT` model, it is not possible to use both states in the joints and the `FlexBeam` component. Since the elasto-kinematic behavior of a twist beam suspension is caused by both linkage compliance and bushing mounts at chassis [8], the linkage model was modified to use bushing mounts at the chassis instead. This added three additional degrees of freedom to each wheel, increasing the number of required states to 24. This can be achieved by using 6 deformations and their derivatives at one bushing as states with the rest of the states from the `FlexBeam` as shown in Figure 3.26.

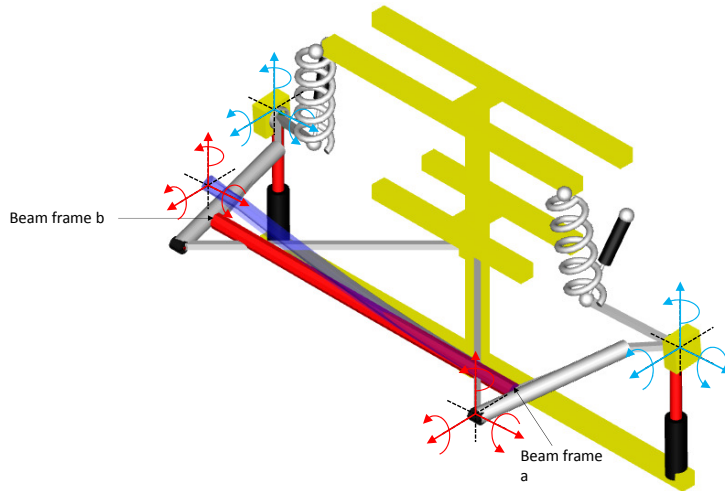


Figure 3.26: *State selection and degrees of freedom in modified `TwistBeamTT` VDL linkage model with chassis mount bushings and `FlexBeam` (10x scaled beam deformation shown in purple).*

Another model using the `FlexBeamBushings` component was also built in VDL which had states enabled

in both the bushings and compliant attachment at either end of the FlexBeam component (in addition to the chassis mount bushings). To further improve the performance of this model, the rotation sequence to rotate the forces and displacements in the beam's local coordinate system was changed from quaternions to Euler angles. This got rid of the non linear system of equations in the beam model. However, it was found that the simulation time was dependent on the beam attachment bushings and the deformations at the wheel started deviating further from the FlexBeam model for larger wheel movements.

Two types of simulations were performed using the hub rig for evaluating the performance of these variants:

- Kinematics test (36 sec, 500 steps).
- Compliance test (200 sec, 500 steps).

The input for the kinematics test is vertical displacement input: the right hub is held at a fixed increments while the left hub is swept through the entire range of wheel travel as shown in Figure 3.27. In the compliance test, the hub is fixed in the vertical position and forces and torques are applied to each wheel in turn as shown in Figure 3.28.

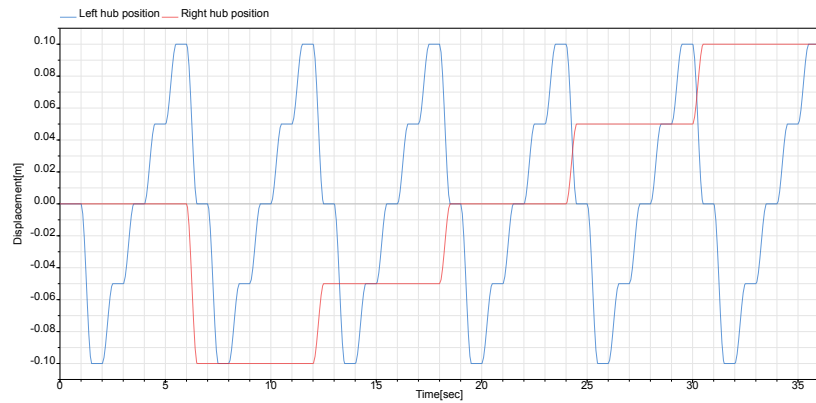


Figure 3.27: *Hub relative vertical displacement input for kinematics test.*

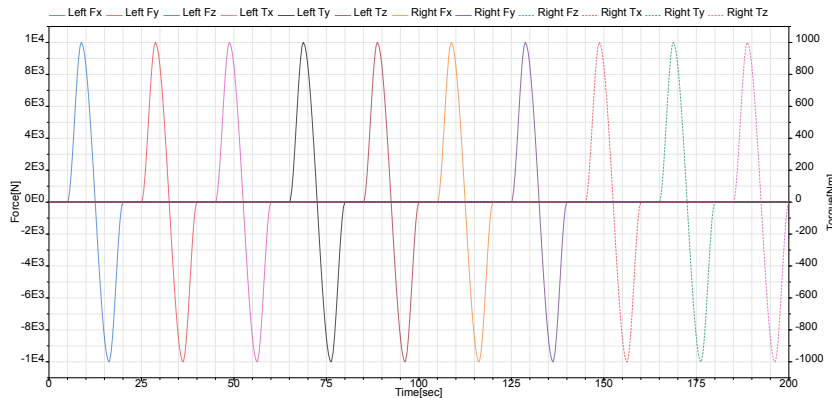


Figure 3.28: *Hub force and torque input for compliance test (Vehicle coordinate system).*

The default Dassl solver was used with the default tolerance settings and a circular cross section with a diameter of 0.025m was selected in the FlexBeam. The tested variants are summarized in Table 3.6. The beam animation was also turned off for these tests.

Table 3.6: Beam model variants tested in hub rig.

| Beam model                                               | FlexBeam      | FlexBeam-Bushing | FlexBeam-Bushing | FlexBeam-Bushing      |
|----------------------------------------------------------|---------------|------------------|------------------|-----------------------|
| Nomenclature                                             | FB            | FBB1             | FBB2             | FBB3                  |
| FlexBeam attachment bushing translation stiffness (N/m)  | -             | $9 * 10^{10}$    | $1.5 * 10^7$     | $1.5 * 10^7$          |
| FlexBeam attachment bushing rotation stiffness(Nm/rad)   | -             | $8 * 10^{10}$    | $8 * 10^8$       | $8 * 10^8$            |
| FlexBeam attachment bushing translation damping(Nsec/m)  | -             | $3 * 10^9$       | $1 * 10^3$       | $1 * 10^3$            |
| FlexBeam attachment bushing rotation damping (Nmsec/rad) | -             | $4 * 10^9$       | $4 * 10^3$       | $4 * 10^3$            |
| Rotation sequence                                        | Quar-ternions | Quar-ternions    | Quar-ternions    | {1,2,3}(Euler angles) |

Appendix D shows example translation and simulation logs from one of the simulations. The parameters used for performance evaluation as part of this thesis are the following:

- Sizes of linear systems of equations: It shows the number of systems of linear equations and the number of variables in each equation (e.g. {3,3,3,3} represents four systems of equations with three variables each).
- Sizes of nonlinear systems of equations: It uses the same notation but represents the nonlinear systems of equations.
- CPU time for integration: Shows the CPU time required for simulation of the model.

Since Dymola uses symbolic reduction of DAEs, the sizes of the systems of equations can also be seen after the substitution and elimination of variables in the translation log under sizes after manipulation of the linear/nonlinear systems [12]. Lower sizes of systems of equations (and lesser number of variables) provide better simulation performance as do linear systems compared to nonlinear systems.

For further improvement in simulation time (valid for most models and recommended for real time model translation) evaluation of all parameters in a model can be turned on. By default, Dymola evaluates only structural parameters (i.e. parameters required to run a model). With this flag (Evaluate:=true) turned on all parameters except top level parameters are evaluated to generate more efficient code [12, 25]. This can be seen in the translation log of a given model by checking the sizes of the linear and non-linear systems of equations after translation. The CPU time is obtained from the simulation logs and provides an indication of computational cost of simulating a Dymola model. An example simulation log is shown in Appendix D.2.

Since one of the advantages of the FlexBeam component is being able to turn on the respective bending and torsion dynamic mode shape as required, the kinematics and compliance tests were also performed after turning on different combinations of dynamic mode shapes as shown in Table 4.6. Though these tests are supposed to be quasi-static and should not excite the eigenfrequency of the beam, observation on the performance impact of dynamic mode shapes can be made. The nomenclature used for denoting number of dynamic mode shapes is {Dynamic mode shapes in axial direction, dynamic mode shapes in Y bending direction, dynamic mode shapes in Z bending direction, dynamic mode shapes in torsion}. For example {1,1,1,1} denotes one dynamic mode in each direction.

### 3.4 Cantilever beam test model

In order to ensure that the correct cross section properties were used in both ABAQUS and Dymola and the translation to ADAMS MNF was done correctly, a cantilever beam model of 1m length and the cross section properties from Table 3.4 was setup in all 3 software (ADAMS/View was used to evaluate the MNF). The static deformation and natural frequency values were compared using the methods shown in Table 3.7.

Table 3.7: Simulation methods used for testing cantilever beam.

| Software   | Method for static deformation        | Method for natural frequency                              |
|------------|--------------------------------------|-----------------------------------------------------------|
| ABAQUS     | Linear static analysis               | Normal modes analysis                                     |
| ADAMS/View | Static simulation                    | ADAMS/Linear                                              |
| Dymola     | Dynamic simulation with ramped force | Time period measurement after setting initial deformation |

The respective details on the methods used can be found in [10, 17, 31]. The loads and boundary conditions for the static deformation test in the Dymola diagram layer are shown in Figure 3.29. Mode shapes from ABAQUS and ADAMS for the first bending mode about Z axis are shown in Figure 3.30.

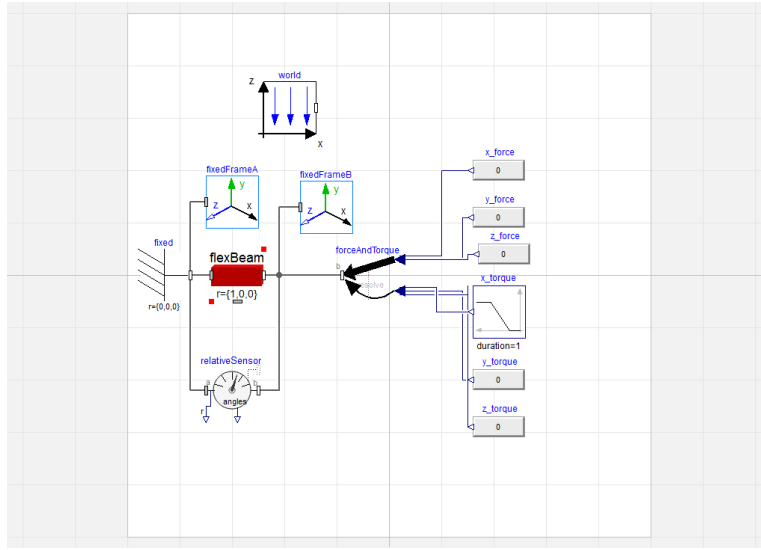
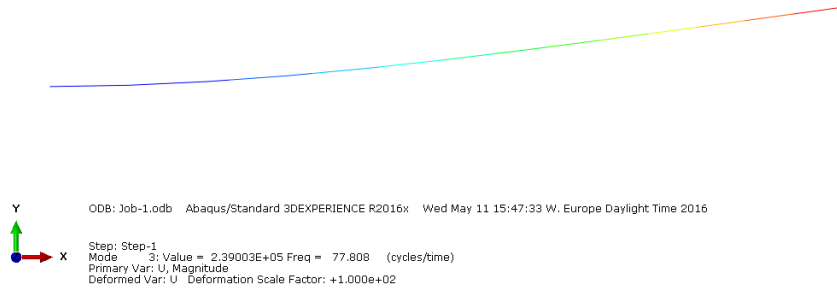
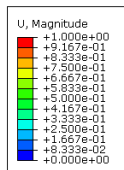
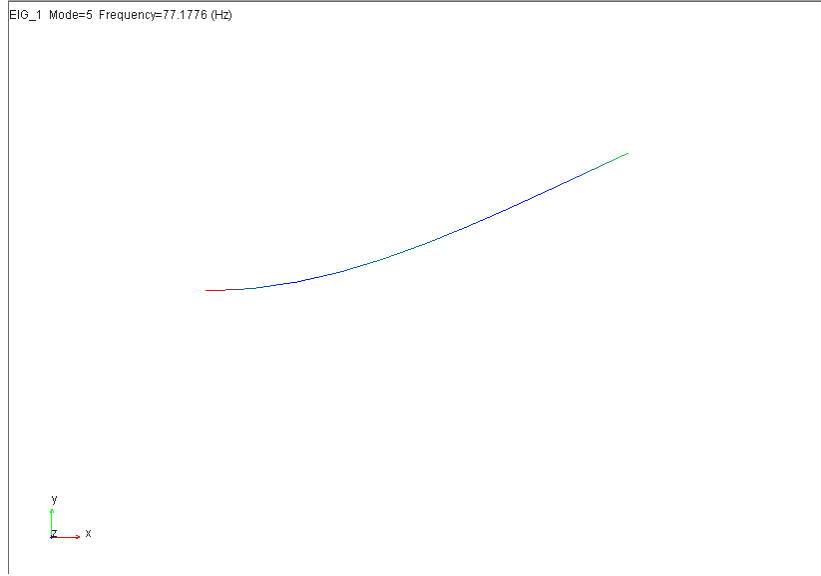


Figure 3.29: Diagram layer view of cantilever beam test model in Dymola.



(a) ABAQUS/CAE displacement plot.



(b) *ADAMS/View displacement plot.*

Figure 3.30: *Mode shape for first bending mode about Z axis.*

### 3.5 Kinematics and Compliance tests in VDL and ADAMS/Car

The input provided to the testrig for the kinematics test in both ADAMS/Car and VDL is shown in Table 3.8. All displacement inputs are relative to the wheel center. This was achieved in ADAMS/Car by selecting "Parallel wheel travel" for the analysis with the same name and "Static loads" for the rest 2 cases. The respective dialog boxes are shown in Figure 3.31. For the roll test, the coordinate system was changed to ISO. In VDL, the same functionality is achieved by turning on position control in Z for the actuator in the wheel rig controller and redeclaring the source to be a ramp function as shown in Figure 3.32.

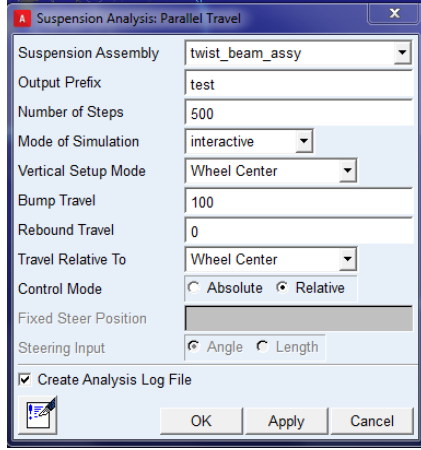
Table 3.8: Displacement inputs for kinematic tests.

| Test                              | Parallel wheel travel | Opposite wheel travel | Roll motion |
|-----------------------------------|-----------------------|-----------------------|-------------|
| Left wheel vertical position(mm)  | 0 to 100              | 0 to 50               | 0 to 50     |
| Right wheel vertical position(mm) | 0 to 100              | 0 to -50              | 0 to -50    |
| Tilting ground plane              | Off                   | Off                   | On          |

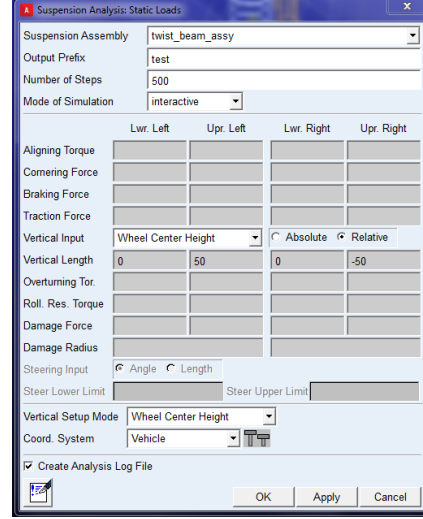
The inputs for the different cases in the compliance tests are shown in Table 3.9. The static loads dialog box was used again in ADAMS/Car with the wheel vertical length set to 0 and the respective cornering, braking force and aligning torque was entered for the different load cases. In VDL, the vertical position control was set to 0 wheel position and the ramp sources were used to input forces at the contact patch and the aligning torque at the hub.

Table 3.9: Force inputs for compliance tests.

| Test                              | Lateral force | Longitudinal force | Aligning torque |
|-----------------------------------|---------------|--------------------|-----------------|
| Left wheel vertical position(mm)  | 0             | 0                  | 0               |
| Right wheel vertical position(mm) | 0             | 0                  | 0               |
| Force X left(N)                   | 0             | 0 to -5000         | 0               |
| Force X right(N)                  | 0             | 0 to -5000         | 0               |
| Force Y left(N)                   | 0 to 5000     | 0                  | 0               |
| Force Y right(N)                  | 0 to 5000     | 0                  | 0               |
| Torque Z left(Nm)                 | 0             | 0                  | 0 to 5000       |
| Torque Z right(Nm)                | 0             | 0                  | 0 to 5000       |

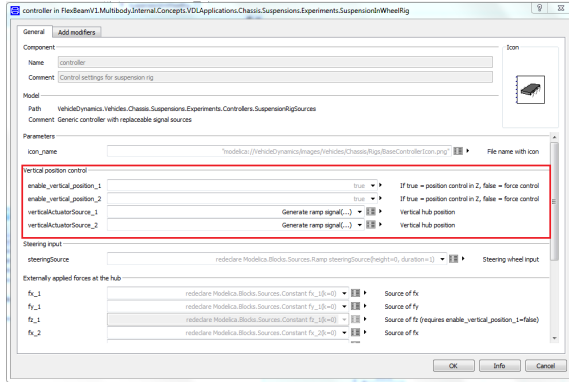


(a) Parallel wheel travel dialog box.

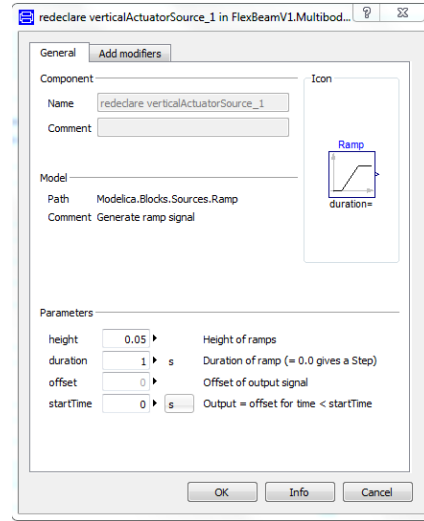


(b) Static loads dialog box.

Figure 3.31: Suspension analysis dialog boxes used in ADAMS/Car.



(a) VDL controller inputs.



(b) Dialog box for ramp source in VDL.

Figure 3.32: VDL wheel rig controller settings and input source.

### 3.6 Chassis Modeling in VDL

In order to evaluate the performance of the FlexBeam component when used in a full vehicle, a chassis model (consisting of front and rear suspension, body and the wheels) was built using an existing template for a compact car in VDL. A compliant double wishbone suspension was used in the front and a twist beam suspension using the FlexBeamME component was used in the rear. The diagram layer view of the chassis model is shown in Figure 3.33. 205/55R17 Pacejka 02 tires are used on all 4 corners of the chassis model. The twist beam suspension uses ADAMS/Car bushings and 2 FlexBeamME components, with the rest of the components and parameters left as default.

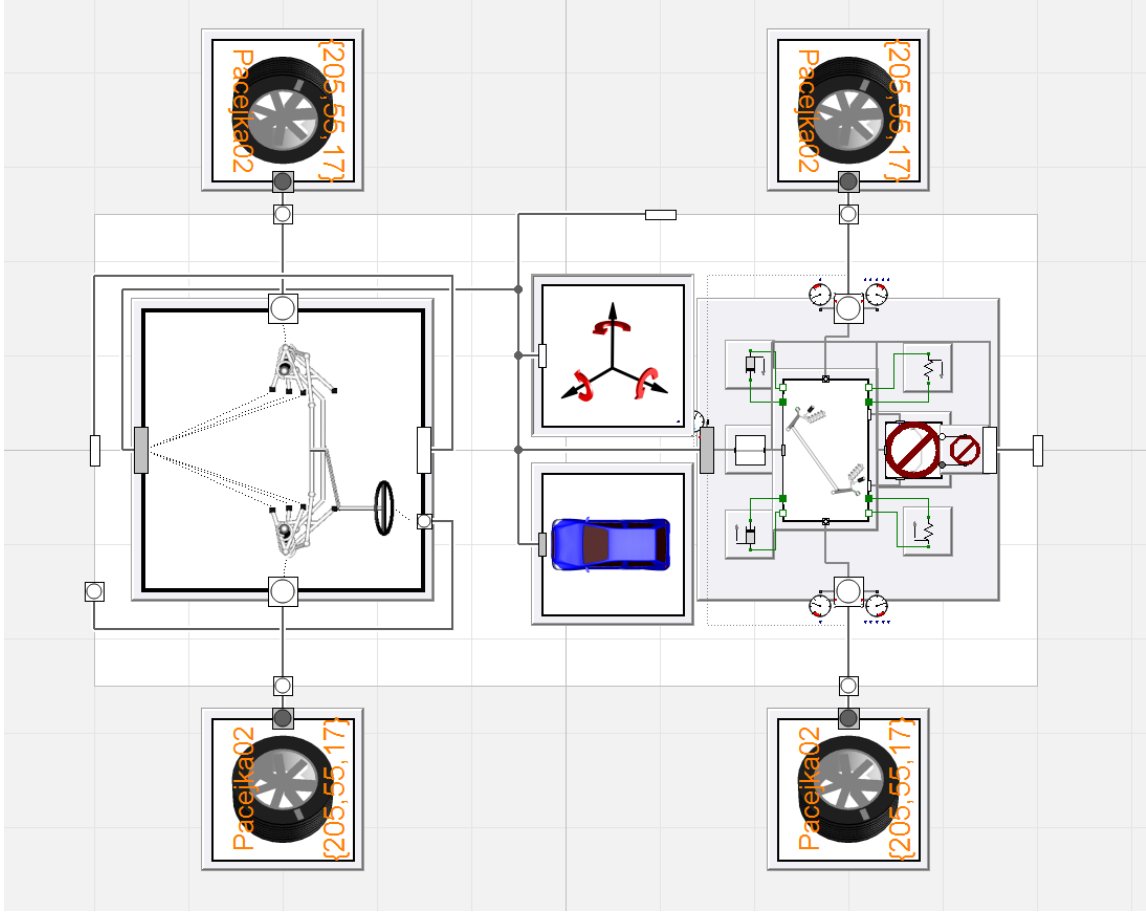


Figure 3.33: *Diagram layer view of chassis model in Dymola using a compliant double wishbone suspension in front and a twist beam rear suspension.*

### 3.6.1 Real time simulation in Dymola

For real time simulation of models, a fixed step solver can be selected in Dymola instead of the default Dassl solver, which is a variable step solver [12]. Inbuilt timing functions also enable the turnaround time (Solver time required for each output increment) for each increment to be monitored and provide information on overruns, i.e. when the turnaround time is larger than the step size. Another feature available in Dymola, is inline integration which is a combined numeric and symbolic approach to solving DAEs [12]. This feature allows generation of efficient code for real time applications. For the real time performance evaluation of the chassis model, the implicit Euler solver was used with a step size of  $0.001\text{sec}$ . The Dassl solver was used for comparison to baseline results. All the simulations were done on a laptop running Windows 7 on a Intel Core i7-3740QM quad core CPU and 8GB RAM.

### 3.6.2 Dymola decouple block

Since implicit methods generate one large system of non-linear equations, a decouple operator in Dymola was introduced which delays the physical signal by one time step [15]. This enables the decomposition of a large system of equations to several smaller systems which can then be executed in parallel. The diagram layer view of the VDL decouple mount component (based on the Dymola block) is shown in Figure 3.34. The user can define the physical quantity to be decoupled: potential or flow corresponding to force or position in mechanical systems.

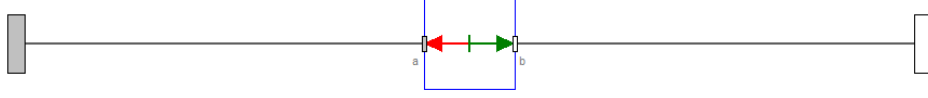
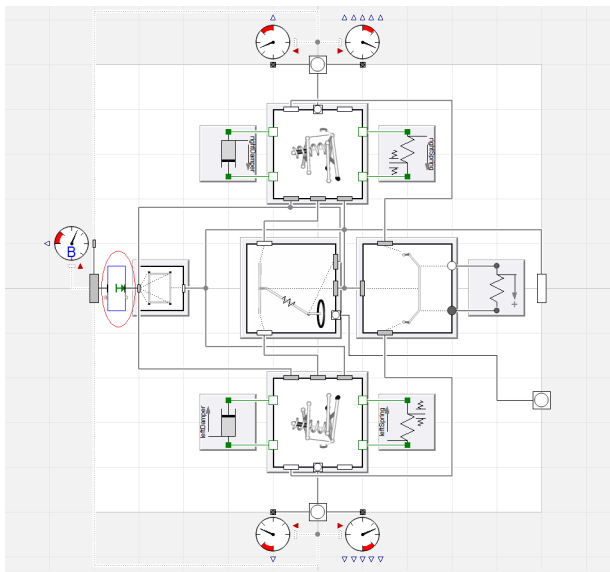


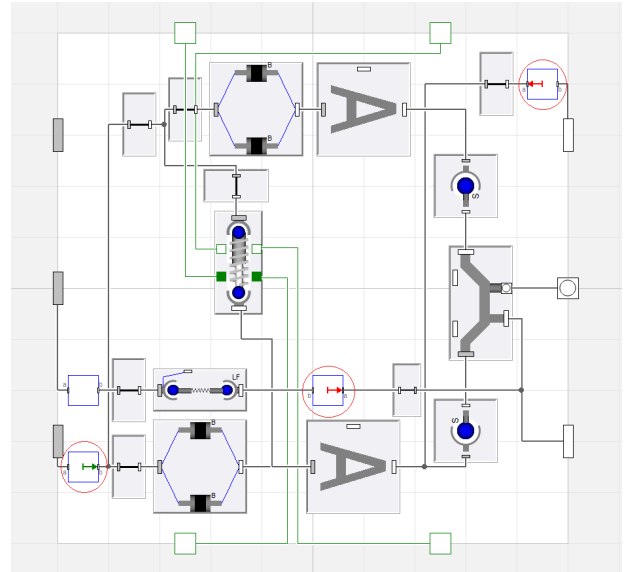
Figure 3.34: *Diagram layer view of VDL decouple mount.*

For running multiple systems of equations from the model in parallel, the front and rear suspension of the chassis were decoupled. The front suspension was a compliant double wishbone suspension from the VDLRealTime package (Package by Modelon, containing real time optimized models and components which are derivatives of VDL components), which features additional decoupling of left and right suspension, steering and anti-roll bar as shown in Figure 3.35. One potential decoupler is used to decouple the rear suspension from the body. Figure 3.36 shows the diagram layer view of the linkage model. To ensure that the solution accuracy is not adversely affected because of the use of this block, comparison to a variable step solver with no decoupling was also done.

Additional information is added to the translation log when a model is executed on multiple cores as shown in Appendix D.3. It shows the CPU operation counts which would be used without parallelization, the number of parallel executions achieved and the operation count for each parallel execution section [12]. To leverage the processing capability available from modern multi-core CPUs, it is desirable to have multiple parallel executions with smaller number of operations rather than a single sequential operation with a large number of operations [15].



(a) *Suspension assembly.*



(b) *Linkage model.*

Figure 3.35: *VDLRealTime compliant double wishbone suspension model with decouple blocks.*

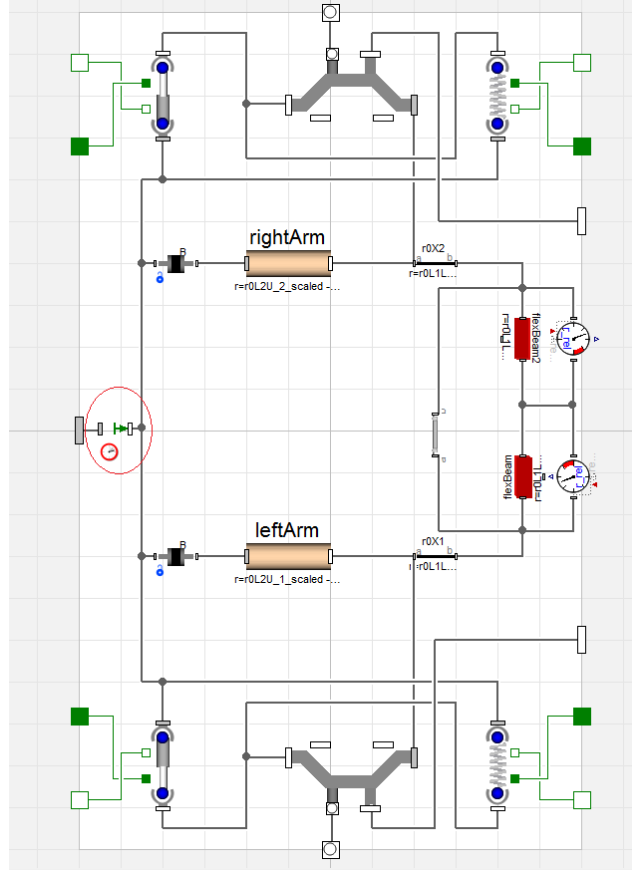
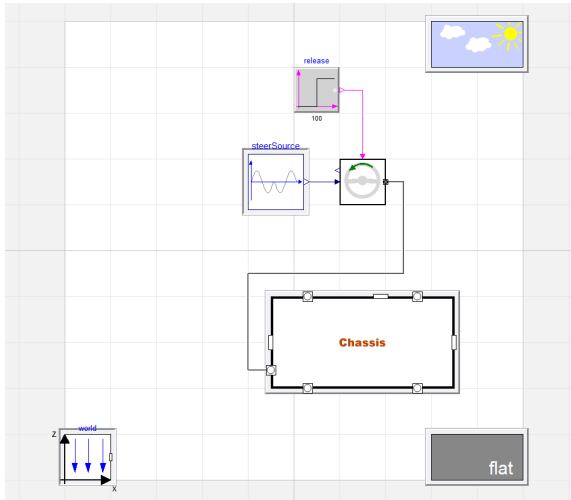


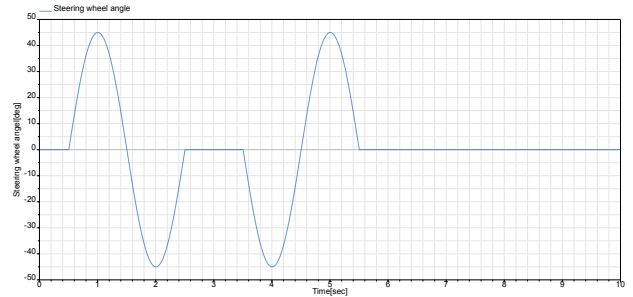
Figure 3.36: Diagram layer view of twist beam rear suspension model with decouple mount.

### 3.6.3 Chassis experiments

Similar to experiments on suspension assemblies, chassis experiments can also be done by providing steering input or specifying the target states of the chassis (position, velocity, acceleration). An open loop, double lane change steer maneuver was used for the evaluation of chassis model for real time performance. Figure 3.37 shows the diagram layer view of the chassis experiment and the steering input with time.



(a) Diagram layer view of chassis experiment.



(b) Steering input.

Figure 3.37: Diagram layer view and steering input of open loop steer chassis experiment.

### 3.6.4 Chassis model variants tested for real time performance improvement

The variants tested in the open loop steer maneuver test are summarized in Table 3.10.

Table 3.10: Chassis model variants tested for real time simulation performance.

| Variant   | Front suspension decoupling                                         | Rear suspension decoupling | Discretization in FlexBeamME | Solver         |
|-----------|---------------------------------------------------------------------|----------------------------|------------------------------|----------------|
| Base-line | None                                                                | None                       | 10 elements                  | Dassl          |
| RT1       | None                                                                | None                       | 10 elements                  | Implicit Euler |
| RT2       | None                                                                | Decoupled from front       | 10 elements                  | Implicit Euler |
| RT3       | Decoupled from rear, left-right, steering, anti roll bar decoupling | Decoupled from front       | 10 elements                  | Implicit Euler |
| RT4       | Decoupled from rear, left-right, steering, anti roll bar decoupling | Decoupled from front       | 2 elements                   | Implicit Euler |
| RT5       | None                                                                | Decoupled from front       | 2 elements                   | Implicit Euler |

The baseline model is used to ensure that the usage of an Implicit Euler solver and the multiple decouplers being used do not significantly affect the simulation results. The variants RT2 and RT3 include the increasing usage of decouplers with RT2 using only front and rear suspension decoupling and RT3 having additional decoupling between front left and front right suspension. Variants RT4 and RT5 have the same decoupling as variant RT3 and RT2 respectively but uses no discretization in the FlexBeamME elements (only two FlexBeam components are used). The results from the open loop steer test for these variants are shown in Section 4.5.

## 4 Results

### 4.1 Component Level Tests on Cantilever Beam

The results from component level tests on the cantilever tests are shown in Table 4.1 and Table 4.2. For the dynamic tests in Dymola, one dynamic shape function was used for the respective directions and the stiffness damping was set to  $1 * 10^{-6}$ . It can be seen that both the FlexBeam and the MNF beam show similar deformation behavior to the finite element ABAQUS beam. The same is the case for Y and Z bending natural frequency. The MNF beam has a higher natural frequency for axial direction while the FlexBeam is closer to the finite element beam.

Table 4.1: Static deformation test results for cantilever beam.

| Direction | Force(N) | Moment(Nm) | ABAQUS                | Dymola                | ADAMS/View             |
|-----------|----------|------------|-----------------------|-----------------------|------------------------|
| X         | 1000     | -          | $5.137 * 10^{-6}$ m   | $5.126 * 10^{-6}$ m   | $5.1 * 10^{-6}$ m      |
| Y         | 1000     | -          | $2.322 * 10^{-3}$ m   | $2.310 * 10^{-3}$ m   | $2.323 * 10^{-3}$ m    |
| Z         | 1000     | -          | $2.398 * 10^{-3}$ m   | $2.338 * 10^{-3}$ m   | $2.398 * 10^{-3}$ m    |
| X         | -        | 100        | $1.993 * 10^{-1}$ rad | $1.990 * 10^{-1}$ rad | $1.989 * 10^{-1}$ rad  |
| Y         | -        | 100        | $7.161 * 10^{-4}$ rad | $8.095 * 10^{-4}$ rad | $7.1558 * 10^{-4}$ rad |
| Z         | -        | 100        | $6.931 * 10^{-4}$ rad | $6.931 * 10^{-4}$ rad | $6.0824 * 10^{-4}$ rad |

Table 4.2: Natural frequency results for cantilever beam.

| Direction | Mode | ABAQUS    | Dymola     | ADAMS/View |
|-----------|------|-----------|------------|------------|
| X         | 1st  | 1278.2 Hz | 1333.33 Hz | 1562.8 Hz  |
| Y         | 1st  | 77.383 Hz | 77.51 Hz   | 77.1776 Hz |
| Z         | 1st  | 75.865 Hz | 75.75 Hz   | 75.946 Hz  |

### 4.2 Performance Evaluation and State Selection

For details on the nomenclature used in this section, Section 3.3.8 is referred.

#### 4.2.1 Effect of parameter evaluation

The results of turning on this flag and the corresponding sizes of equations and CPU time for simulation for 2 variants during the kinematics test are shown in Table 4.3. As can be seen significant improvements in simulation performance can be obtained by evaluating all parameters which reduces the size of equations to be solved. For further testing, this option was left on for all simulations.

Table 4.3: Effect of evaluating all parameters in the kinematics test for two model variants.

| Beam model                                        | FB                  | FBB1                        | FB                  | FBB1                        |
|---------------------------------------------------|---------------------|-----------------------------|---------------------|-----------------------------|
| Evaluate parameters to reduce models              | Off                 | Off                         | On                  | On                          |
| Sizes of linear systems of equations              | {39, 209, 12, 6, 6} | {4, 69, 2, 12, 65, 2, 6, 6} | {21, 154, 12, 6, 6} | {4, 47, 2, 12, 47, 2, 6, 6} |
| Sizes after manipulation of the linear systems    | {3, 18, 3, 0, 0}    | {4, 8, 2, 12, 6, 2, 0, 0}   | {0, 14, 0, 0, 0}    | {4, 0, 0, 12, 0, 0, 0, 0}   |
| Sizes of nonlinear systems of equations           | {39}                | {19}                        | {21}                | {19}                        |
| Sizes after manipulation of the nonlinear systems | {3}                 | {4}                         | {3}                 | {4}                         |
| CPU-time for integration(sec)                     | 4.44                | 30.9                        | 2.71                | 18.1                        |

### 4.2.2 Effect of different FlexBeam components

The CPU time used to run the kinematics and compliance tests on the four variants are shown in Table 4.4. It can be seen that the variant FB is computationally most efficient among the tested variants. The variation of the camber and toe angle for the four variants during the kinematic and compliance tests are shown in Figure 4.1 and Figure 4.2 respectively.

Table 4.4: CPU time for integration comparison for kinematics and compliance tests.

| Variant | Kinematics test(sec) | Compliance test(sec) |
|---------|----------------------|----------------------|
| FB      | 2.71                 | 1.05                 |
| FBB1    | 18.1                 | 19.4                 |
| FBB2    | 6.24                 | 2.24                 |
| FBB3    | 5.21                 | 1.99                 |

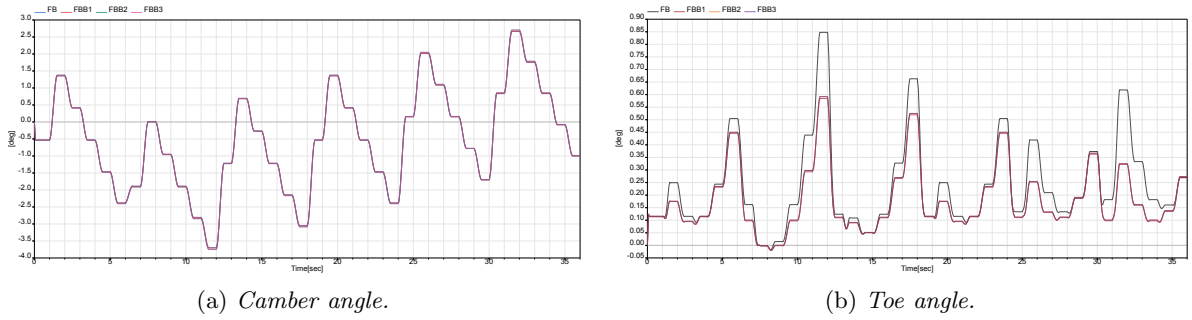


Figure 4.1: Camber and toe angle variation during the kinematics test for different FlexBeam components.

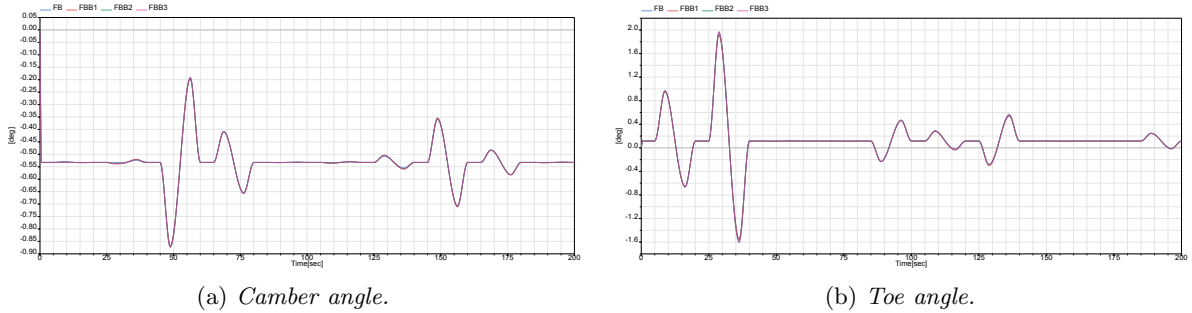


Figure 4.2: Camber and toe angle variation during the compliance test for different FlexBeam components.

It can be seen that the variation in the camber angle is minimal during the kinematics test while the FlexBeamBushing variants exhibit higher toe stiffness with higher deviation from the FlexBeam model at higher relative displacement between the wheel centers. This suggests that the relative orientation of the two FlexBeam bushings affects the toe stiffness. In the compliance test the results are closer for both toe and camber angle as the hubs are held fixed in the vertical direction. The FlexBeamBushing component does have the advantage of being able to be switched out easily in VDL assemblies, without altering the state selection but careful selection of the stiffness and damping of the bushings needs to be done to maintain low simulation times without affecting the simulation results.

### 4.2.3 Effect of turning on dynamic modes in the FlexBeam component

Tables 4.5 and 4.6, show the effect of turning on dynamic modes in the beam for kinematics and compliance test respectively. It can be seen that the variant FB is the most robust and has a somewhat predictable behavior when turning on dynamic modes as the greatest increase in CPU time is when the mode shape with the highest eigenfrequency is turned on.

Table 4.5: Effect of turning on dynamic modes on CPU time for integration(sec) for the four variants in the kinematics test.

| Variant | Dynamic modes-<br>$\{0,0,0,0\}$ | Dynamic modes-<br>$\{1,1,1,1\}$ | Dynamic modes-<br>$\{0,1,1,1\}$ | Dynamic modes-<br>$\{0,3,3,3\}$ | Comments                                                           |
|---------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|--------------------------------------------------------------------|
| FB      | 2.71                            | 319                             | 66.1                            | 52                              | Activating dynamic modes in x largest effect                       |
| FBB1    | 18.1                            | -                               | -                               | -                               | Not simulated with dynamic modes                                   |
| FBB2    | 6.24                            | 92.6                            | 236                             | -                               | Did not run to completion for $\{0,3,3,3\}$ case                   |
| FBB3    | 5.21                            | -                               | 200                             | -                               | Did not run to completion for $\{0,3,3,3\}$ and $\{1,1,1,1\}$ case |

Table 4.6: Effect of turning on dynamic modes on CPU time for integration(sec) for the four variants in the compliance test.

| Variant | Dynamic modes-<br>$\{0,0,0,0\}$ | Dynamic modes-<br>$\{1,1,1,1\}$ | Dynamic modes-<br>$\{0,1,1,1\}$ | Dynamic modes-<br>$\{0,3,3,3\}$ | Comments                                               |
|---------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|--------------------------------------------------------|
| FB      | 1.05                            | 1810                            | 11.9                            | 45.9                            | Activating dynamic modes in x largest effect           |
| FBB1    | 19.4                            | -                               | -                               | -                               | Not simulated with dynamic modes                       |
| FBB2    | 2.24                            | 58.2                            | 882                             | -                               | Did not run to completion for $\{0,3,3,3\}$ case       |
| FBB3    | 1.99                            | -                               | -                               | -                               | Did not run to completion with dynamic modes activated |

Based on the results from these tests, the variant FB was used to validate the VDL model against the ADAMS model. A further modification was made to this model based on the initial tests. It was found that the FlexBeam component exhibited asymmetric behavior when connecting frame a or frame b to the state select bushing. The results showed close agreement to ADAMS results on the side with the state select bushing.

On closer investigation it was found that different deformation values were obtained for the same force on a cantilever beam if frame b was fixed instead of frame a. A difference in the number of equations needed to be solved by Dymola for the two cases could also be seen. A possible reason is the difference in the beam reference coordinate system as ADAMS does not use one end of the beam as the reference [28] (Instead using the origin of the FE model which is symmetric to both ends) while the Dymola beam does have one of the beam ends as reference. As a workaround, two FlexBeamME components were used in the VDL model, each with 5 elements and both frame a connections made with the bushings as shown in Figure 4.3. The results for one output variable, wheel center longitudinal movement from the kinematics test are shown in Figure 4.4. The state selection was left the same as in Section 3.3.8 (States in one bushing and generalized deformation coordinates). The different variants shown in this plot are the following:

- ADAMS: ADAMS/Car model.
- Dymola frame a: VDL model with state select on left side bushing (connected to beam frame a).
- Dymola frame b: VDL model with state select on right side bushing (connected to beam frame b).
- Dymola 2 FBME: VDL model with state select on left side bushing and 2 FlexBeamME components.

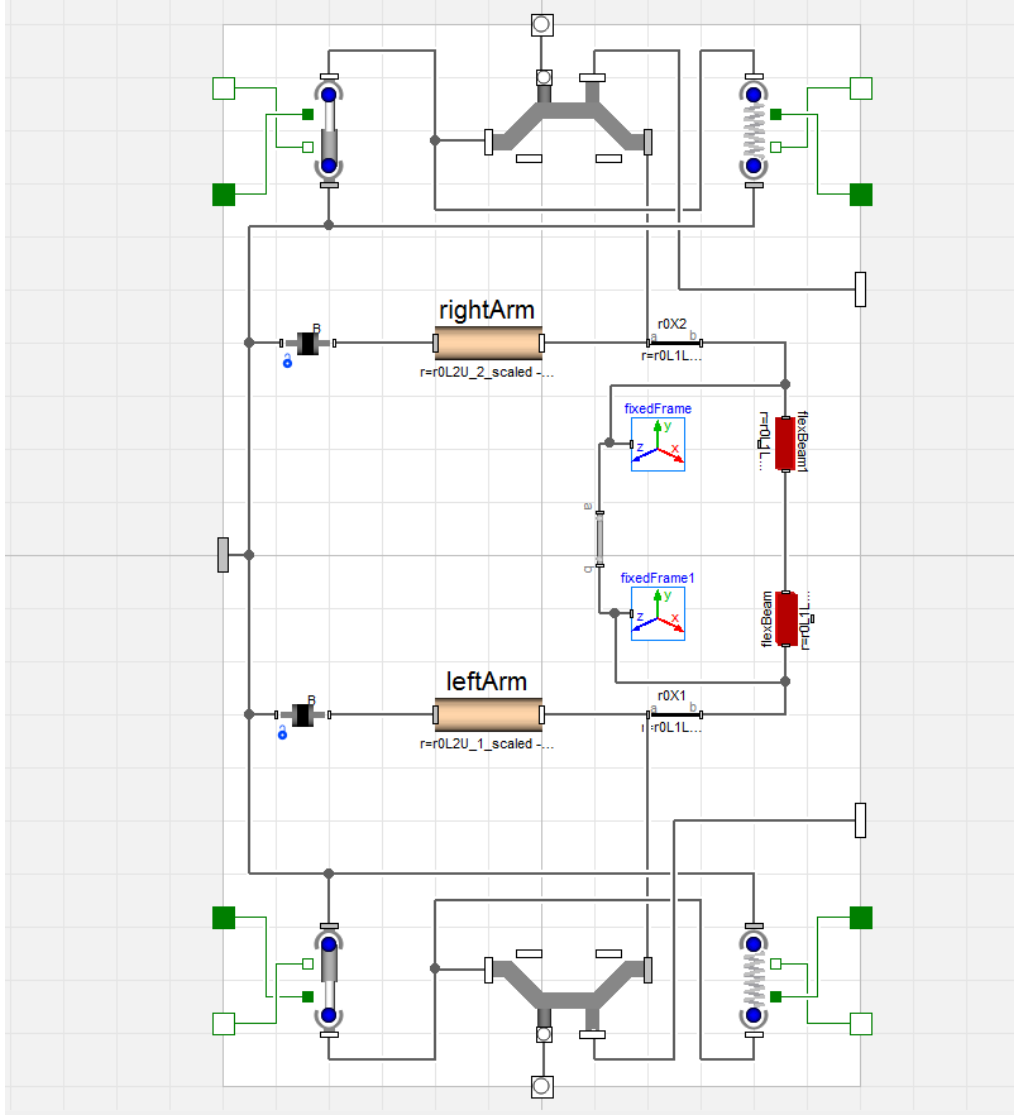


Figure 4.3: *Diagram layer view of the model used for validation against the ADAMS/Car model, using 2 FlexBeamME components with frame a connected to bushing on both sides.*

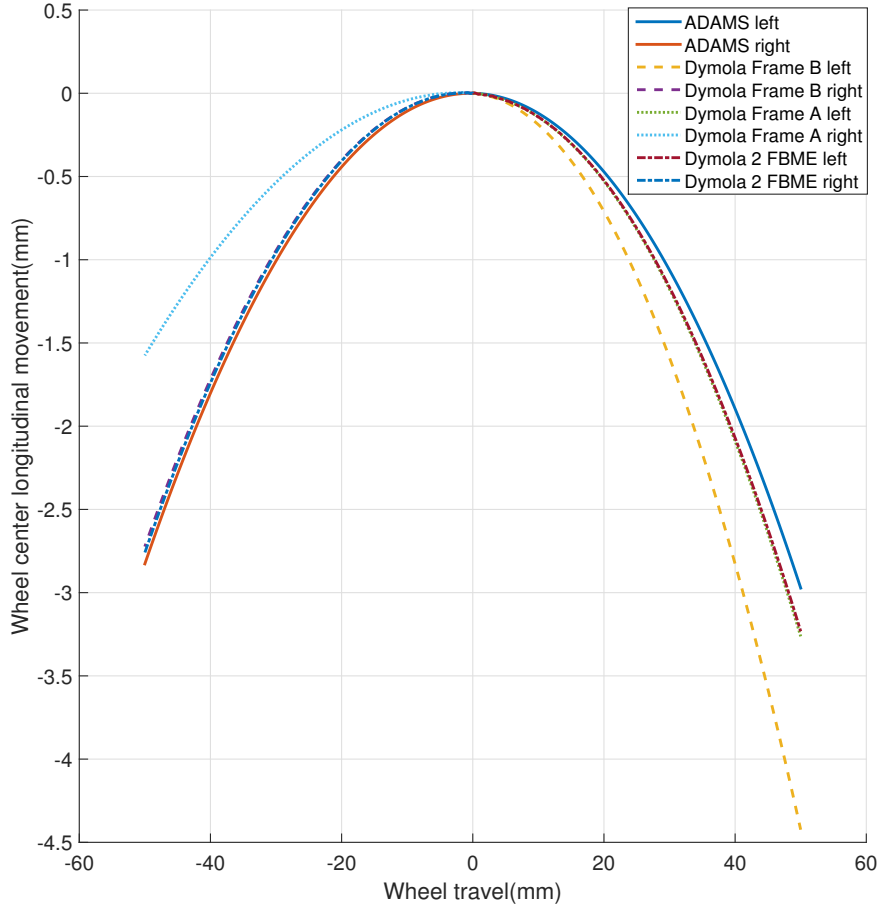


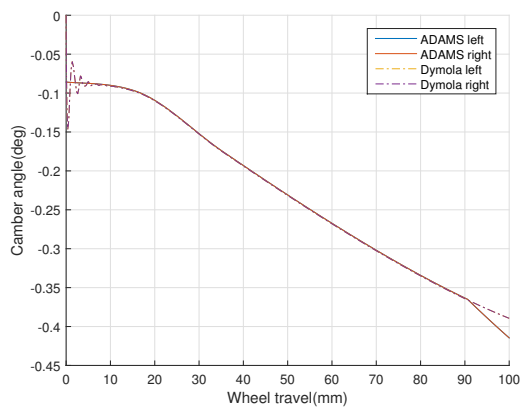
Figure 4.4: *Wheel center longitudinal movement comparison for models with state select on different bushings and with two FlexBeamME components(1 element in each FlexBeamME).*

Closer investigation of the beam formulation and Dymola solution process for the beam is required to get rid of this asymmetry in the beam component.

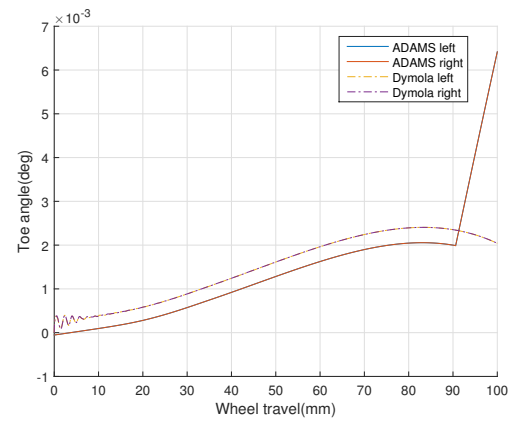
## 4.3 Suspension Kinematics Analysis

### 4.3.1 Parallel wheel travel

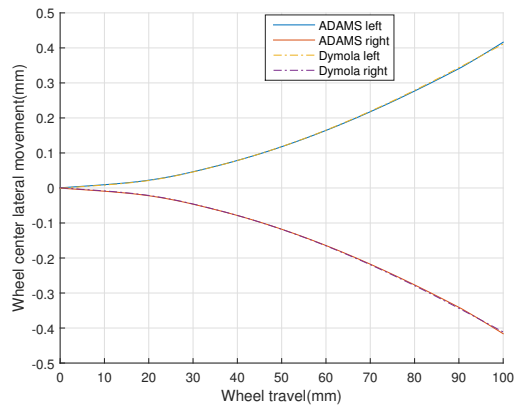
The results from the parallel wheel travel tests are shown in Figure 4.5. It can be seen that the VDL results are in close agreement with most variables from ADAMS/Car, with some deviation at higher wheel travel(90.6mm) when the bump stop is activated in ADAMS. There is also a static offset present in VDL toe angle as shown in Figure 4.5b. The bushing forces are also in close agreement with the Y and Z forces interchanged in VDL due to difference in coordinate systems (ADAMS/Car forces are in bushing coordinate system). Non linear behavior can be seen in most variables from both models caused by the non linear spring (specifically for forces) as well as the kinematics (wheel center movement) . Initial transients are present in the VDL results as the simulation is not quasi-static unlike ADAMS/Car. These can be minimized by offsetting the input ramp signal, instead of starting at  $t = 0$ . Figure 4.6 shows the variation of right chassis mount forces with time in VDL when the simulation is run over 20sec and the ramp signal is applied from 5 to 10sec. The same behavior was seen in the rest of the output variables.



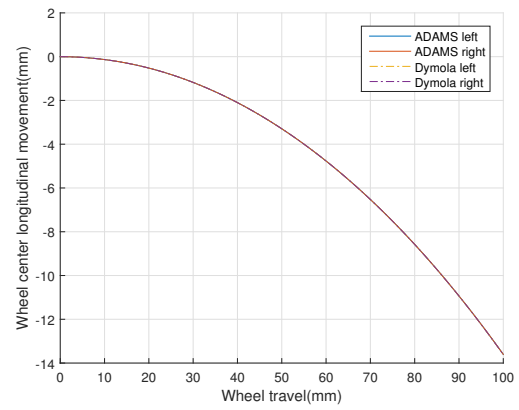
(a) *Camber angle.*



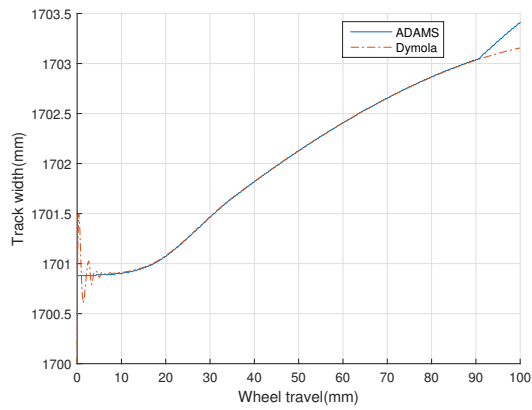
(b) *Toe angle.*



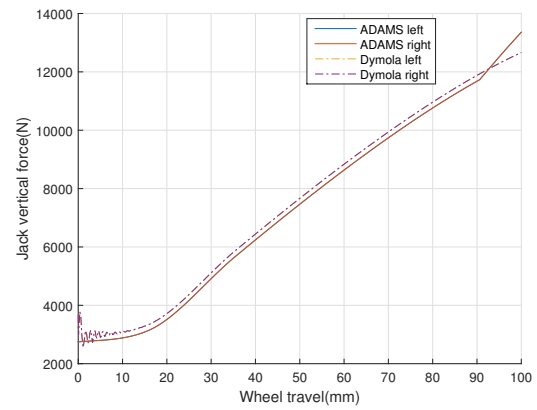
(c) *Wheel center lateral movement.*



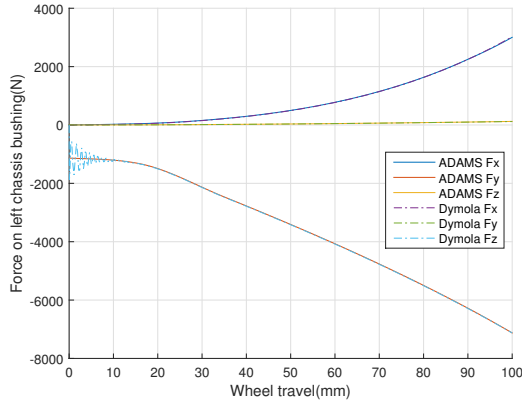
(d) *Wheel center longitudinal movement.*



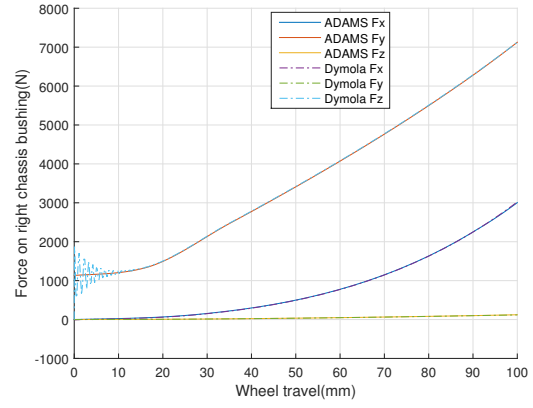
(e) *Track width.*



(f) *Jack vertical force.*



(g) Force on left chassis mount.



(h) Force on right chassis mount.

Figure 4.5: Simulation results of parallel wheel travel tests.

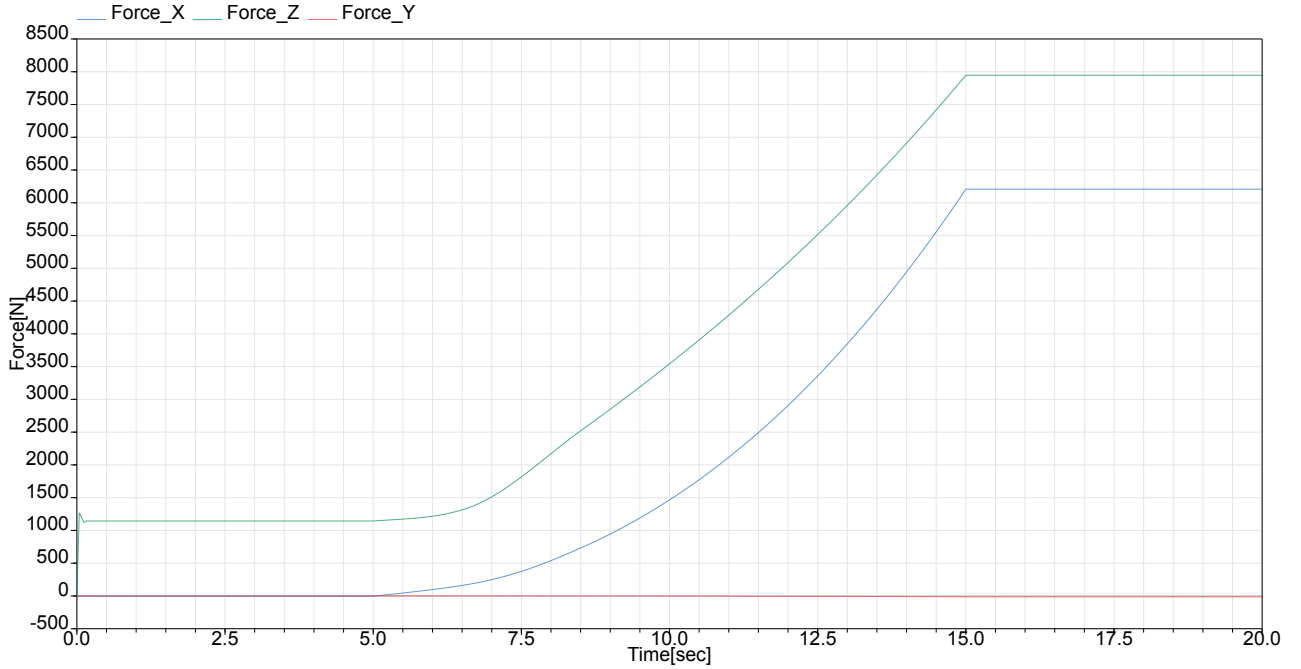
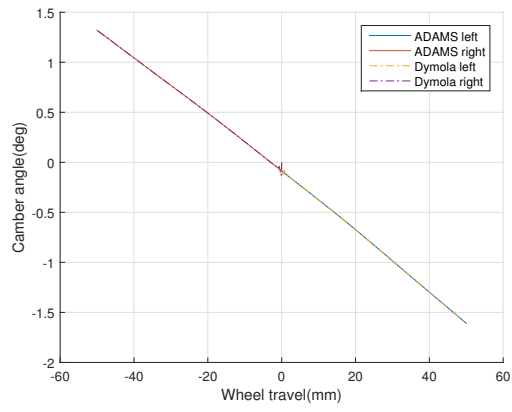


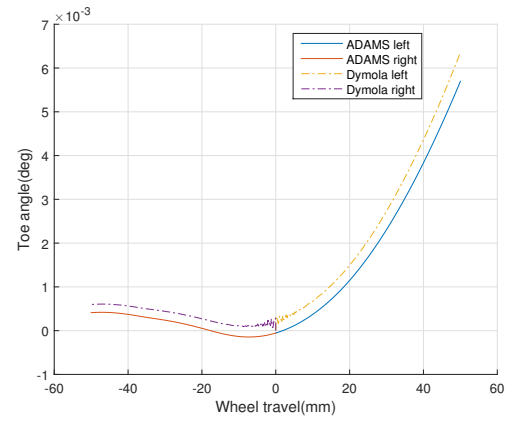
Figure 4.6: Force on right chassis mount bushing in Dymola with offset input signal.

### 4.3.2 Opposite wheel travel

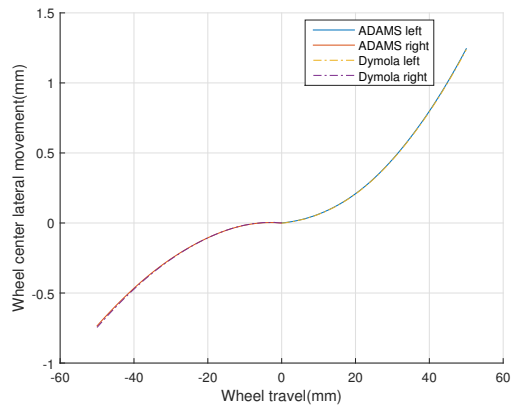
The results from the opposite wheel travel tests are shown in Figure 4.7. A static offset as in the parallel wheel travel tests is present in the toe angle in the VDL results. Additionally, the beam end locations were tracked as well in these tests as shown in Figure 4.8 (The relative displacement between the wheel centers is minimal in the rest of the simulations). The results show that the deformation only due to the beam is quite close for both models in terms of absolute displacements (For the right beam end, the difference in Y direction is around  $0.06\text{mm}$  at  $50\text{mm}$  wheel travel). Similar transients are encountered at the beginning and can be gotten rid of as shown in Figure 4.6.



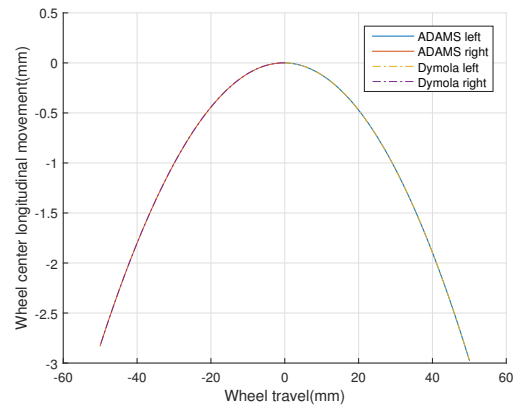
(a) *Camber angle.*



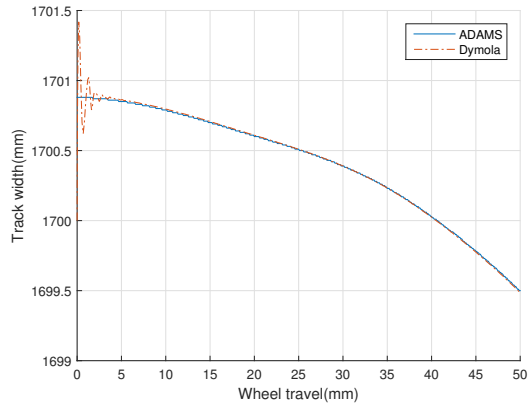
(b) *Toe angle.*



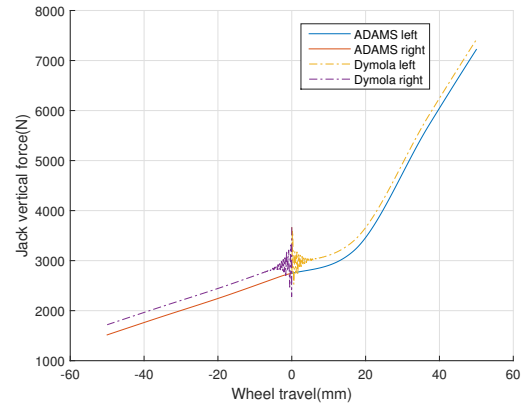
(c) *Wheel center lateral movement.*



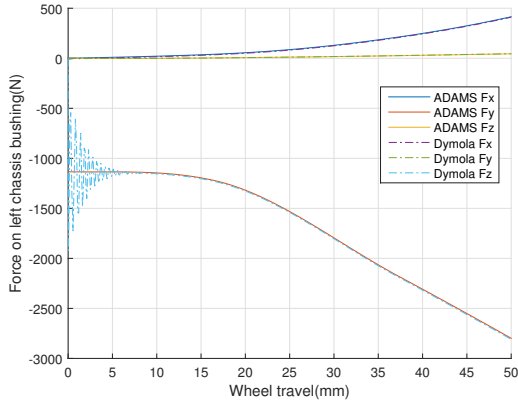
(d) *Wheel center longitudinal movement.*



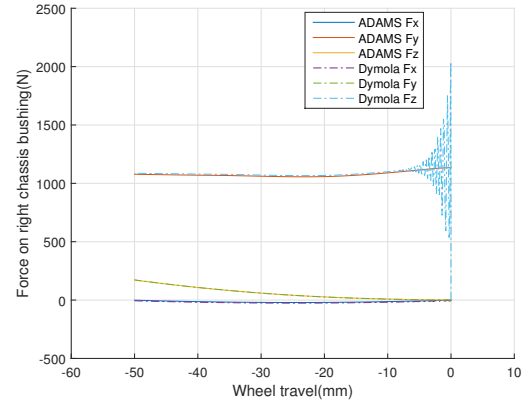
(e) *Track width.*



(f) *Jack vertical force.*



(g) Force on left chassis mount.



(h) Force on right chassis mount.

Figure 4.7: Simulation results of opposite wheel travel tests.

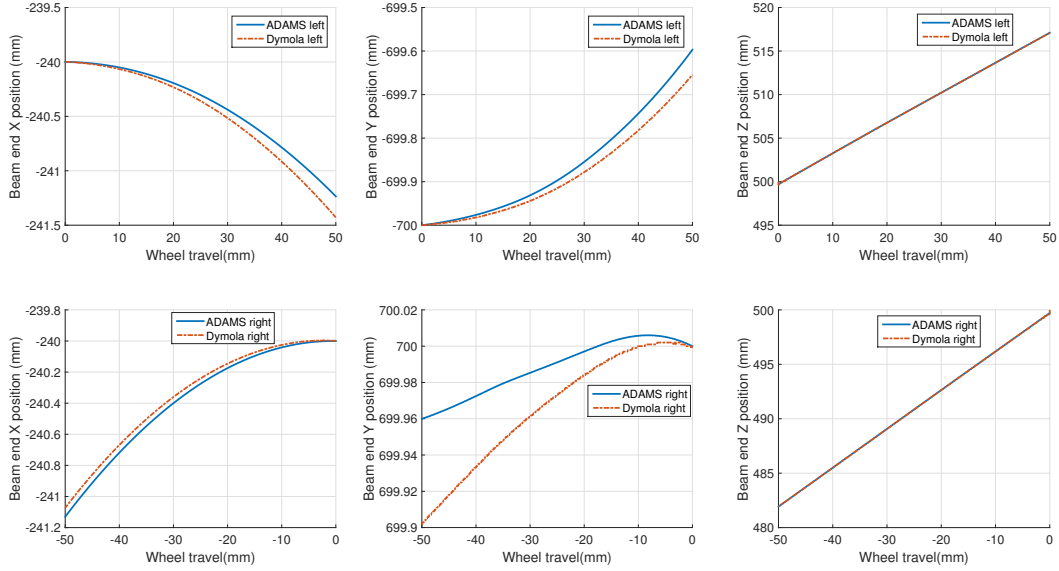
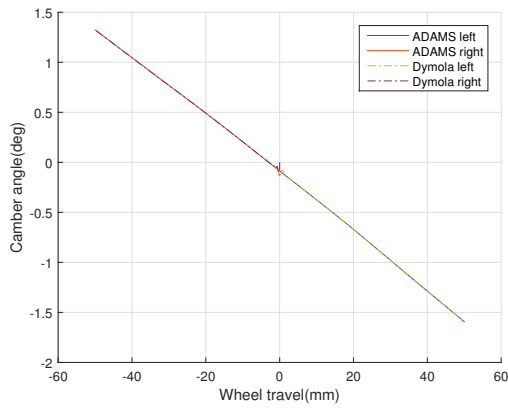


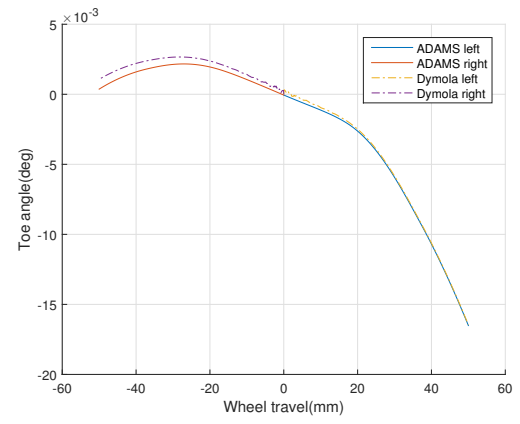
Figure 4.8: Beam end position during opposite wheel travel test.

### 4.3.3 Roll motion

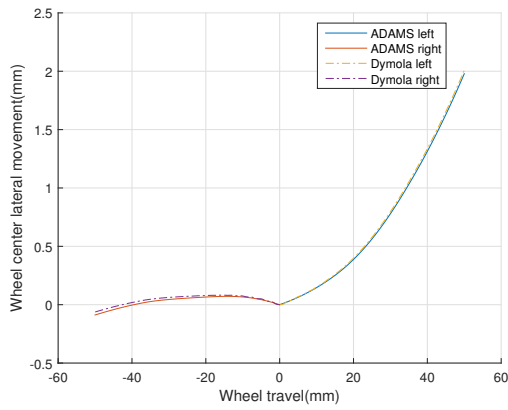
The results from the roll motion tests are shown in Figure 4.9. The primary variable of interest is the toe angle whose variation with roll angle provides a measure of roll steer. It can be seen that the VDL results show good agreement with the ADAMS/Car results. Interestingly, the offset in the toe angle decreases for bump travel and increases for rebound travel between the models. Both models have 0 toe angle at static position. Transients similar to the other tests are seen in the beginning for all variables in VDL.



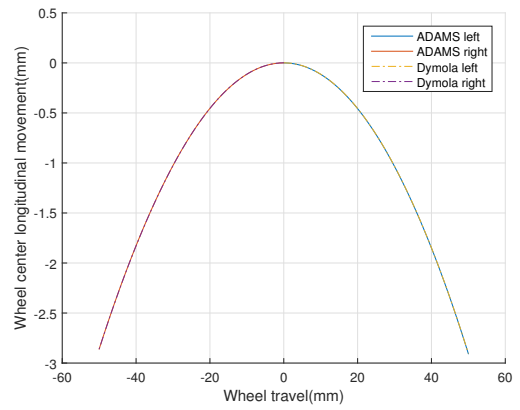
(a) *Camber angle.*



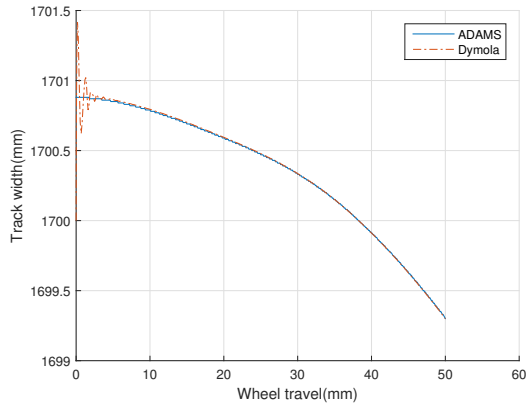
(b) *Toe angle.*



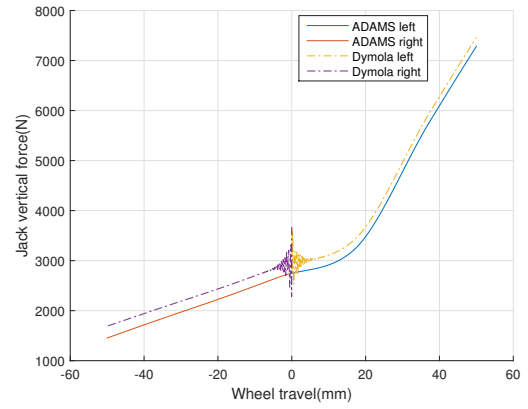
(c) *Wheel center lateral movement.*



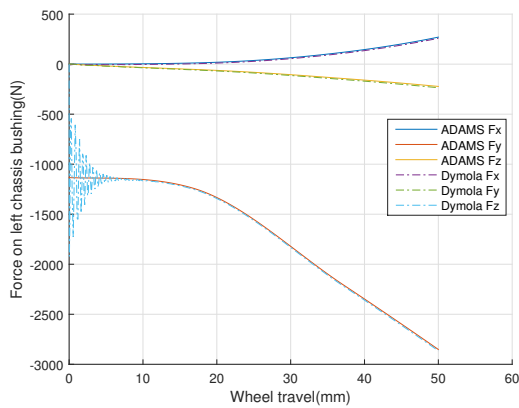
(d) *Wheel center longitudinal movement.*



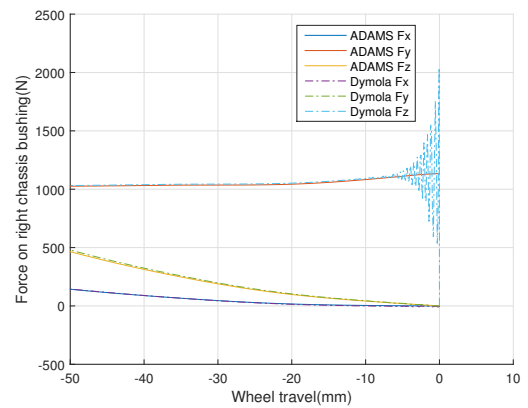
(e) *Track width.*



(f) *Jack vertical force.*



(g) *Force on left chassis mount.*



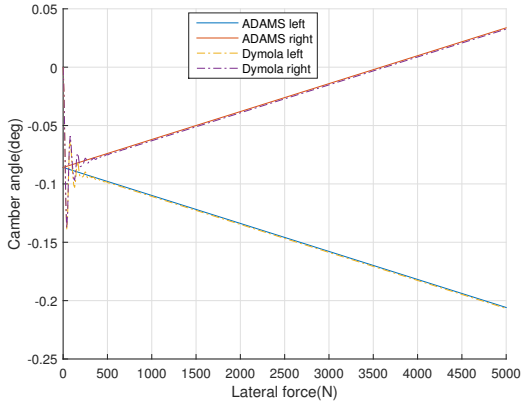
(h) *Force on right chassis mount.*

Figure 4.9: *Simulation results of roll motion tests.*

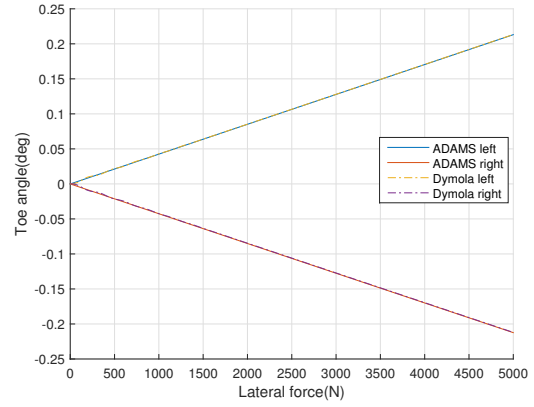
## 4.4 Suspension Compliance Analysis

### 4.4.1 Lateral force compliance

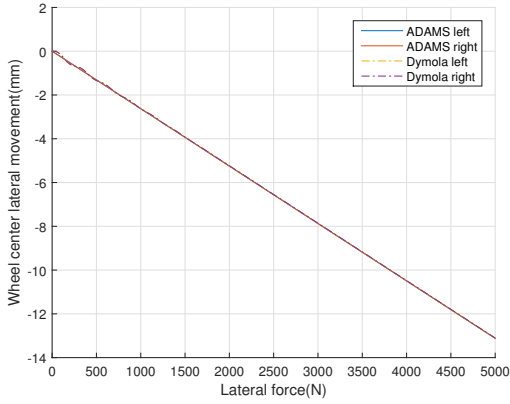
The results from the lateral force compliance test are shown in Figure 4.10. As can be seen there is close agreement for all the variables from both models. There is no offset in the toe angle as was seen in the kinematics test possibly because of the initialization of the wheel centers at zero wheel travel being maintained throughout the simulation and only bending deformation occurring in the beam. Significant variables from this test are camber and toe stiffness to lateral force as well as the wheel center lateral stiffness. The results show linear behavior for the entire magnitude of the applied force as is to be expected from the models.



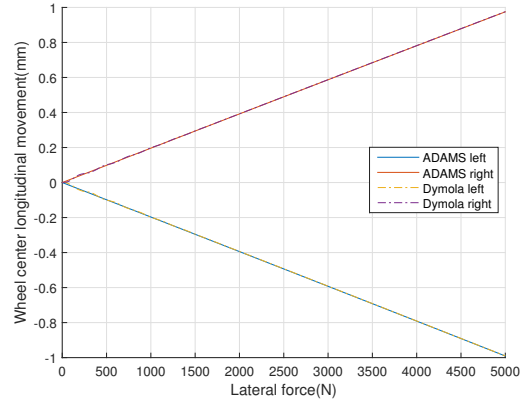
(a) *Camber angle.*



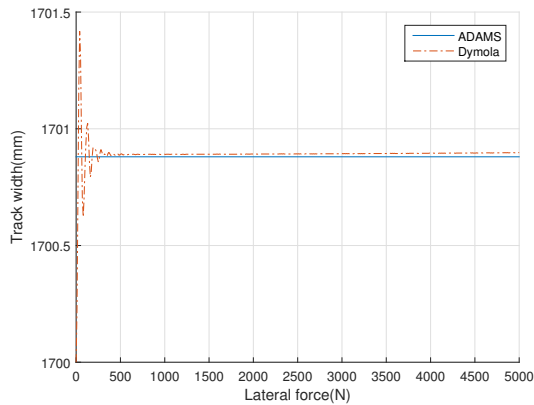
(b) *Toe angle.*



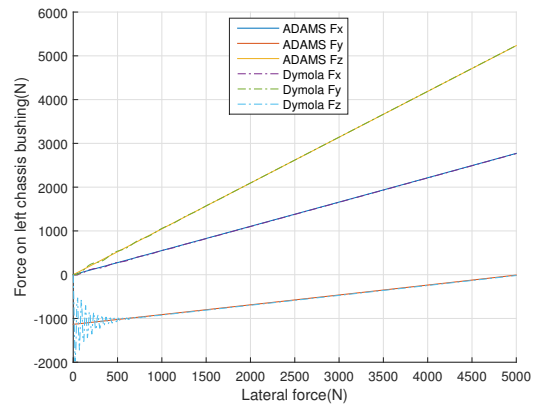
(c) *Wheel center lateral movement.*



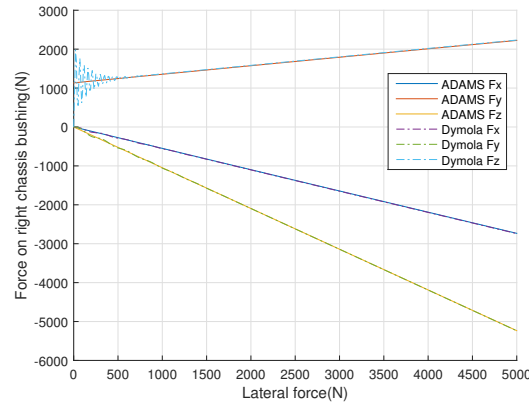
(d) *Wheel center longitudinal movement.*



(e) *Track width.*



(f) *Force on left chassis mount.*

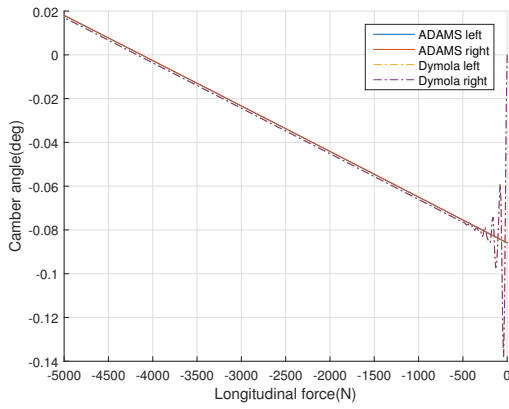


(g) Force on right chassis mount.

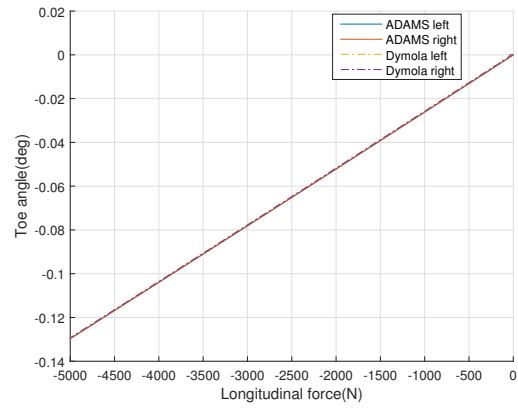
Figure 4.10: Simulation results of lateral force compliance test

#### 4.4.2 Longitudinal force compliance

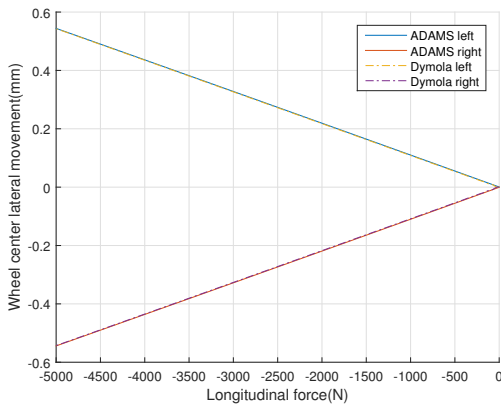
The results from the longitudinal force compliance test are shown in Figure 4.11. As can be seen there is close agreement for all the variables including toe angle from both models. Significant variables from this test are longitudinal force toe and camber compliance as well as the wheel center longitudinal stiffness. The linear behavior is seen for all the variables as seen in the lateral force compliance tests.



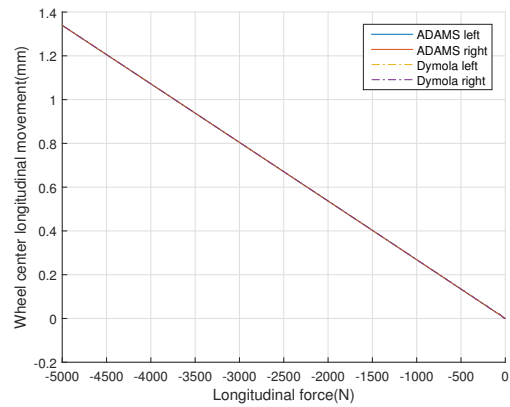
(a) Camber angle.



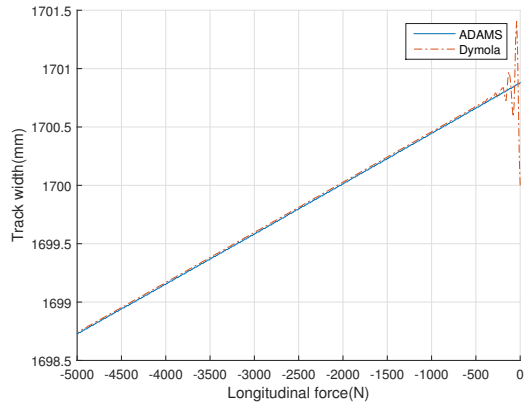
(b) Toe angle.



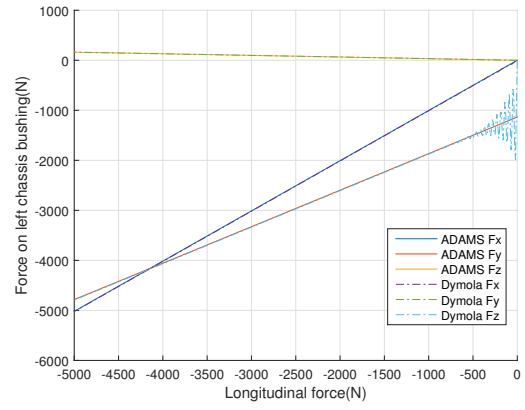
(c) Wheel center lateral movement.



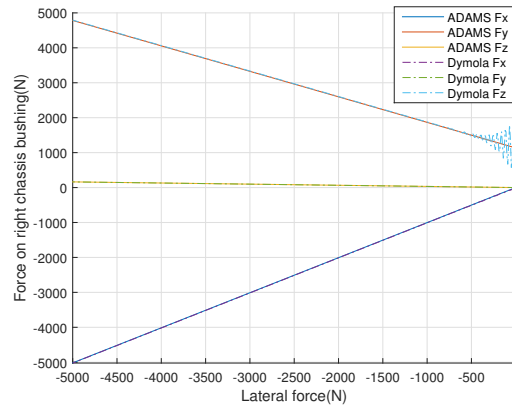
(d) Wheel center longitudinal movement.



(e) Track width.



(f) Force on left chassis mount.

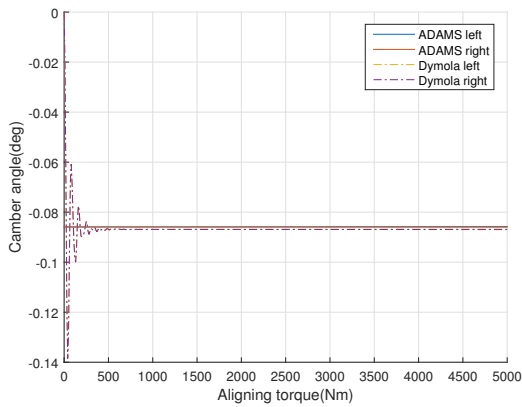


(g) Force on right chassis mount.

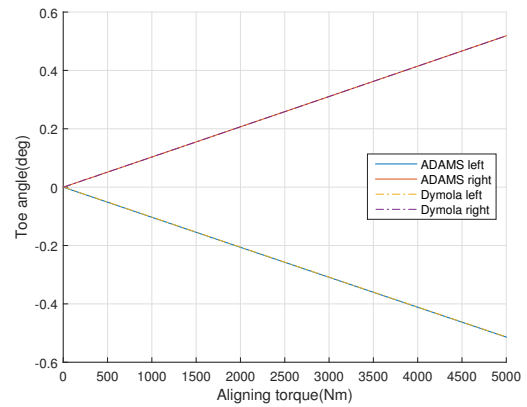
Figure 4.11: Simulation results of longitudinal force compliance test.

### 4.4.3 Aligning torque compliance

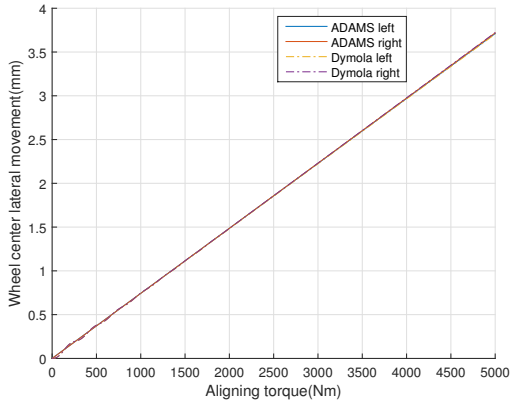
The results from the aligning torque compliance test are shown in Figure 4.12. Close agreement for all variables is seen similar to the rest of the compliance tests. Significant variables from this test are aligning torque toe and camber compliance. The linear behavior is seen for all the variables as seen in the rest of compliance tests.



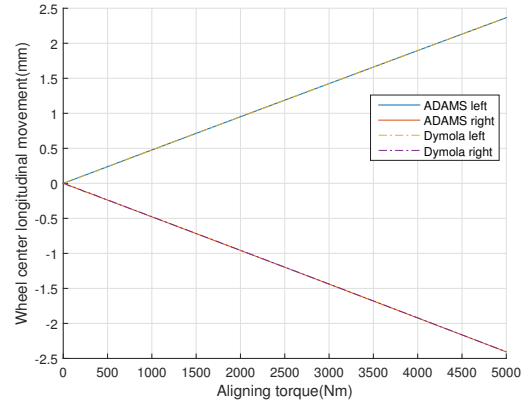
(a) Camber angle.



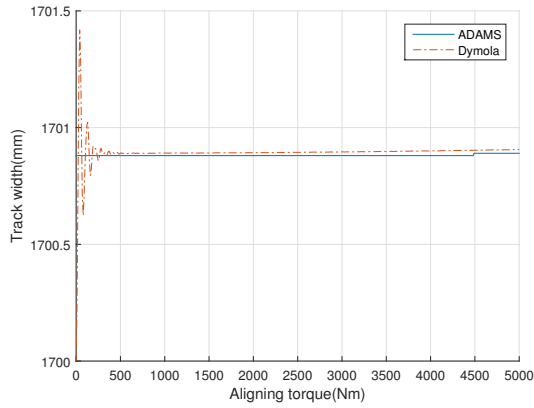
(b) Toe angle.



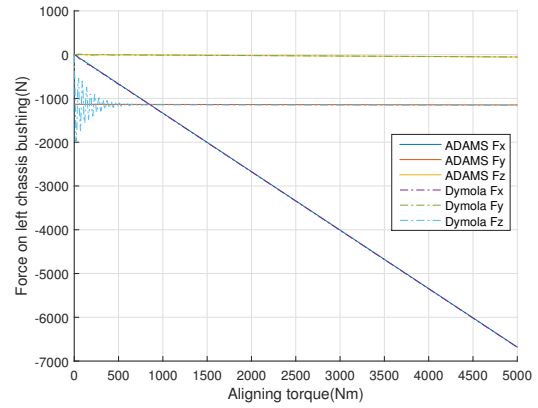
(c) Wheel center lateral movement.



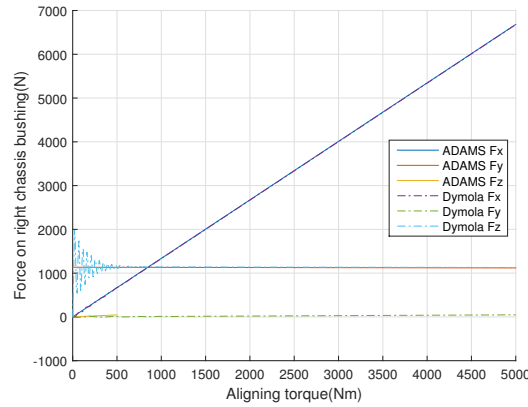
(d) Wheel center longitudinal movement.



(e) Track width.



(f) Force on left chassis mount.



(g) Force on right chassis mount.

Figure 4.12: Simulation results of aligning torque compliance test.

## 4.5 Chassis Model

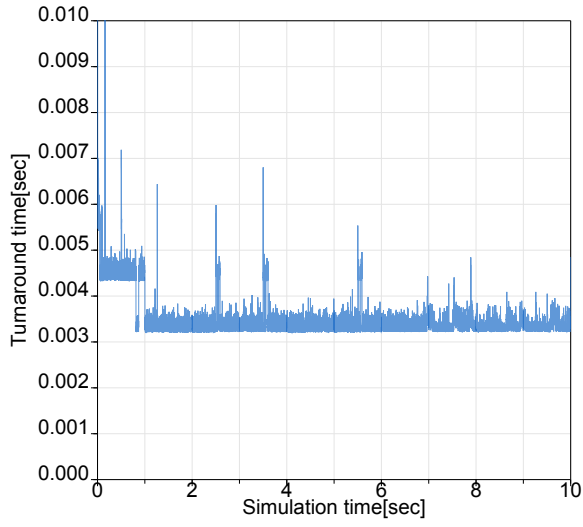
### 4.5.1 Open loop steering test

As mentioned in Section 3.6, the open loop steer test was run using a fixed step solver and the turn around time was monitored and the results were compared to the Dassl solver. The overall CPU time required for the models with the Implicit Euler solver is shown in Table 4.7. The variation of turnaround time with simulation time is shown in Figure 4.13. Figure 4.14 shows the animation view of the chassis during different time frames

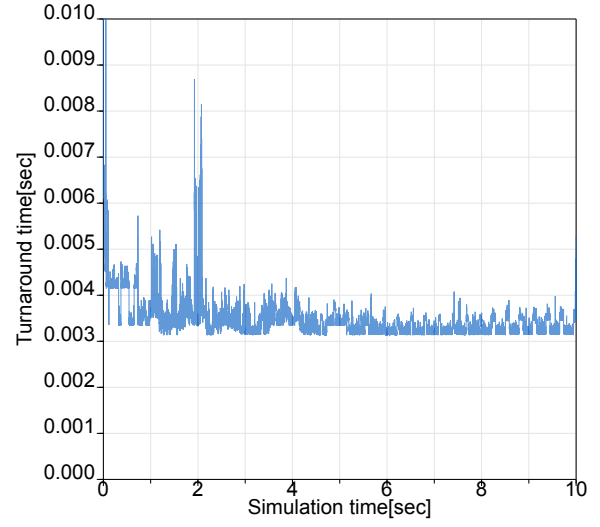
for the baseline model

Table 4.7: Comparison of sizes of equation and CPU times for 4 variants in open loop steer maneuver.

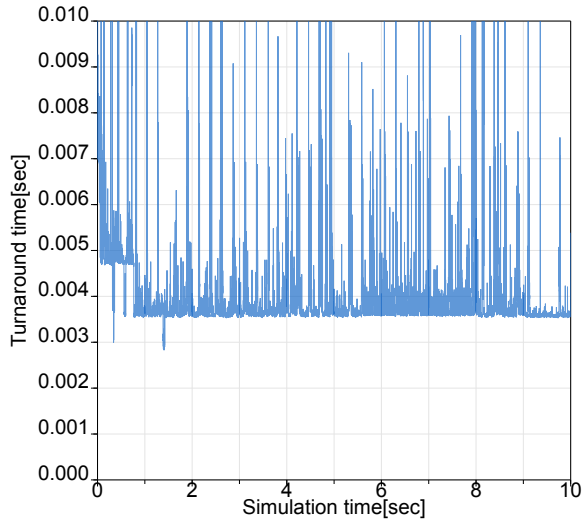
| Variant | Sizes of manipulated inlined implicit integration systems: | CPU time(sec) | Parallelization                  |
|---------|------------------------------------------------------------|---------------|----------------------------------|
| RT1     | {138}                                                      | 38.5          | None                             |
| RT2     | {72, 66}                                                   | 36.8          | None                             |
| RT3     | {72, 23, 23, 14, 6}                                        | 43.6          | 2 Sequential, 1 Parallel section |
| RT4     | {24, 23, 23, 14, 6}                                        | 11.4          | 2 Sequential, 1 Parallel section |
| RT5     | {24, 66}                                                   | 12.2          | None                             |



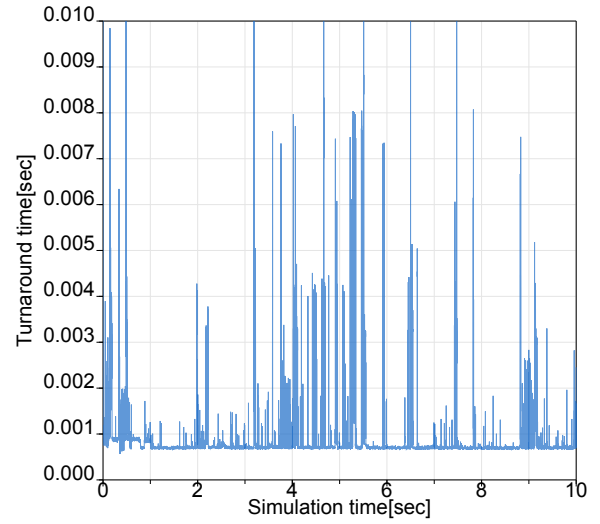
(a) Variant RT1.



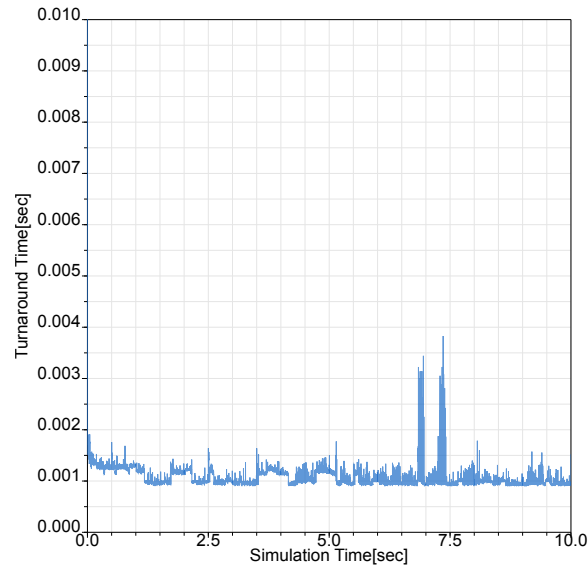
(b) Variant RT2.



(c) Variant RT3.

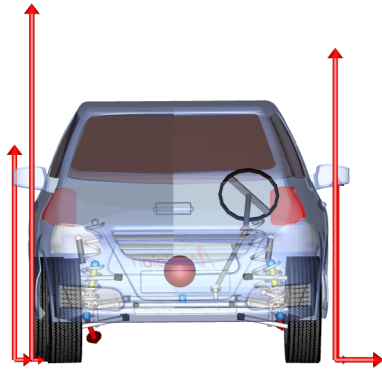


(d) Variant RT4.

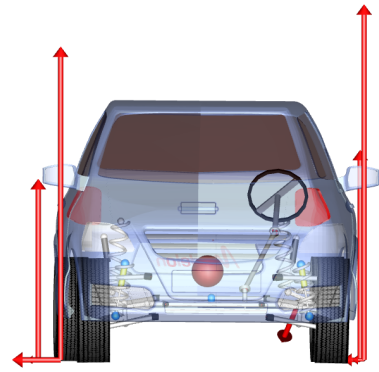


(e) Variant RT5.

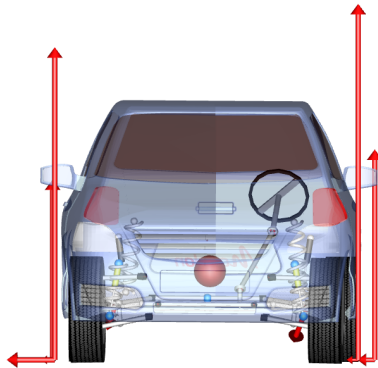
Figure 4.13: Turnaround time for the tested variants.



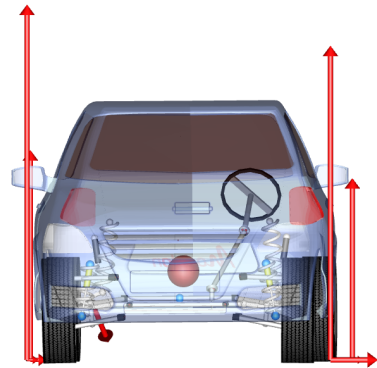
(a)  $t = 1\text{sec}$



(b)  $t = 2\text{sec.}$



(c)  $t = 4\text{sec.}$



(d)  $t = 5\text{sec.}$

Figure 4.14: Animation of the chassis model during different time frames in the open loop steer test.

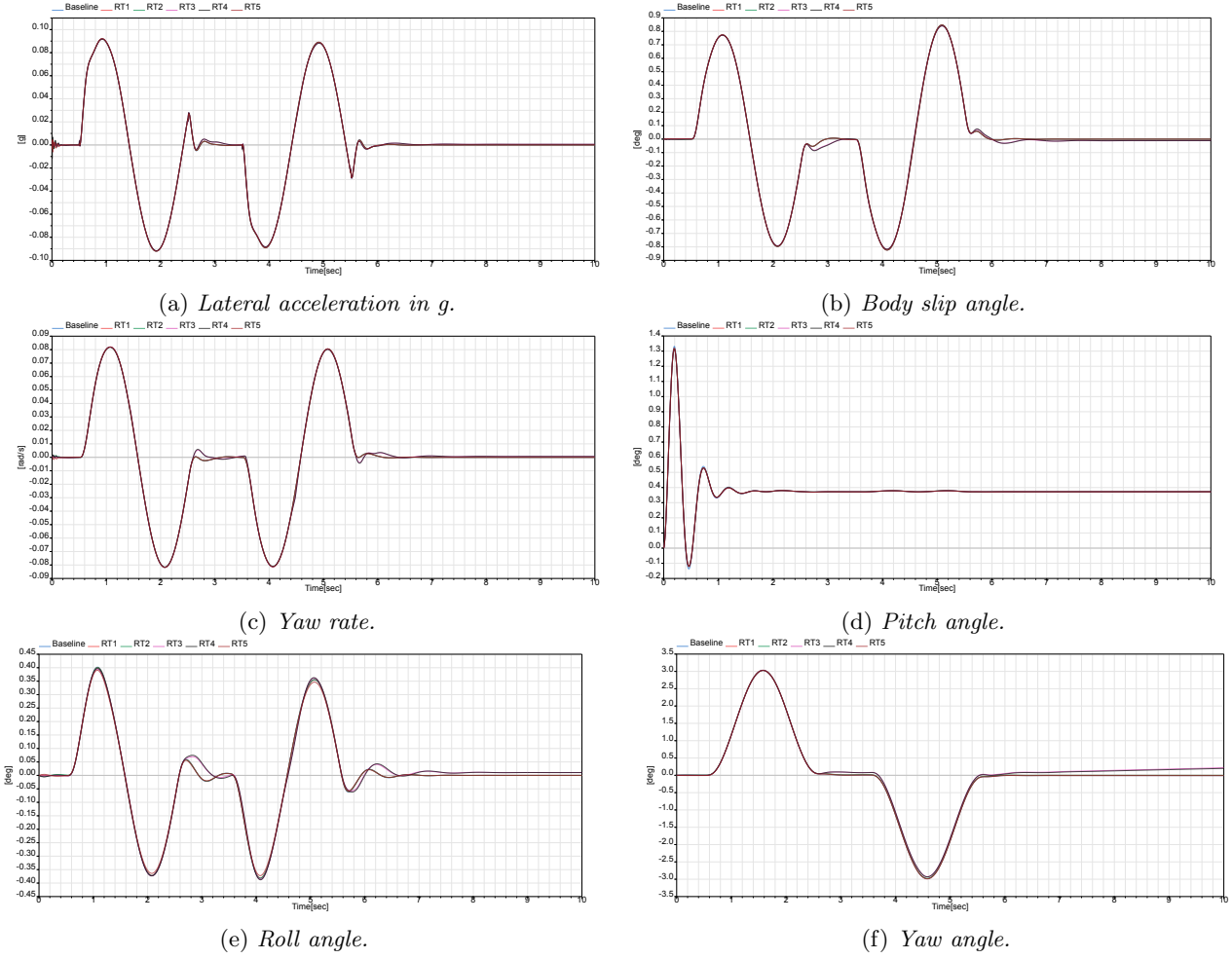
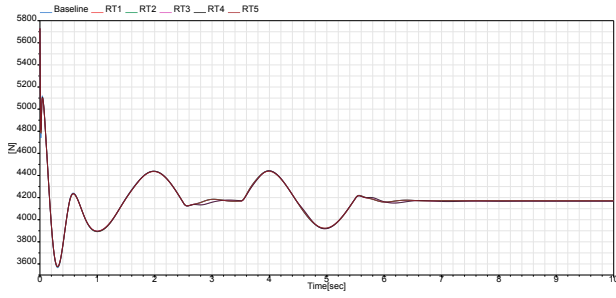


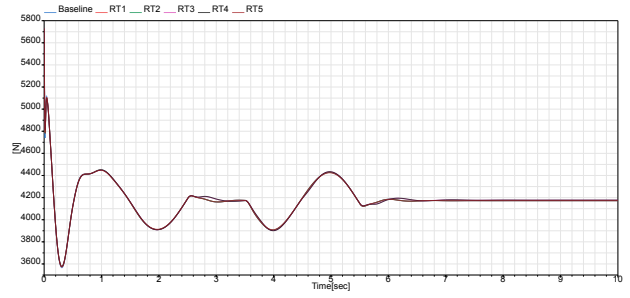
Figure 4.15: Chassis measurement comparison of the five variants during the open loop steer test.

It can be seen that decoupling of the front and rear suspension provides an improvement in the turnaround time while the number of elements used in the FlexBeamME component has a major effect on the turnaround time. For variant RT4, the turnaround time is less than  $0.001sec$  for the majority of the time steps, but some overruns are present. The sources of the overruns were not investigated further as part of this thesis. Interestingly, no parallelization is observed till the front left and right suspensions are decoupled. It can also be seen that there are performance benefits even with no parallelization as the decouplers split the system into two smaller systems of equation in variant RT2 compared to RT1. However, the decoupling of the front left and right suspensions doesn't give performance benefits despite parallelization. Based on the results from RT2 and RT4, variant RT5 was simulated and it has a turnaround time of less than  $0.001sec$  with the maximum turnaround time being less than  $0.002sec$  for the majority of the time steps. Closer investigation of the front and rear suspension and code profiling [12] is required to find out the time step limiting component. Some possible reasons of no performance improvement when the front left and right suspension are decoupled could be that the FlexBeam component is still the CPU time limiting component and mismatched roll stiffness values for the front and rear axles (Default spring and anti roll bar stiffness values were used and one axle might have a much higher percentage of total roll stiffness). A more realistic weight and roll stiffness distribution for a compact car from [6] can be used instead.

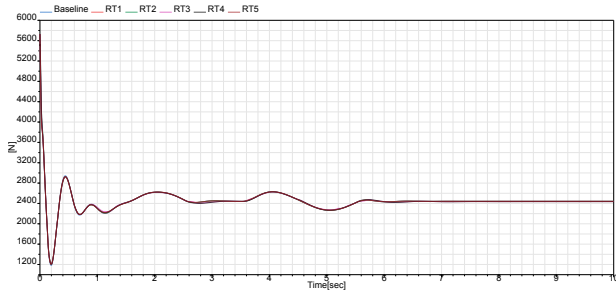
The comparison of some chassis output variables between the five variants and the baseline model is shown in Figure 4.15. It can be seen that the results are in close agreement with the baseline model, even with no discretization in the FlexBeamME component, suggesting that the beam deformation for this specific maneuver and chassis properties (Front and rear roll stiffness, center of gravity height) is within the linear region. The vertical load comparison on all four wheels for these variants is shown in Figure 4.16. This shows that all the variants tested exhibit similar lateral and longitudinal load transfer for this maneuver.



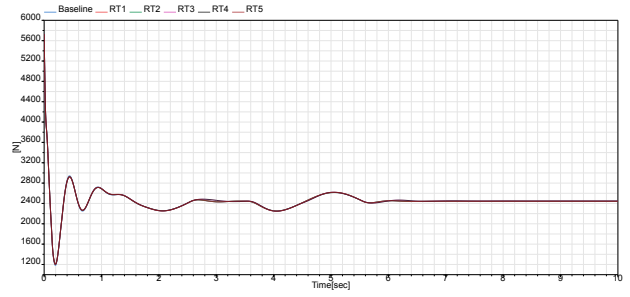
(a) Front left.



(b) Front right.



(c) Rear left.



(d) Rear right.

Figure 4.16: Wheel vertical load comparison of the five variants during the open loop steer test.

## 5 Conclusion

As part of this thesis the following objectives were met successfully:

- The flexible beam model was validated against a finite element beam model and a reduced modal model using ADAMS MNF using a general cross section
- The modifications required for using the beam model in an existing VDL model were investigated and implemented in the form of a twist beam suspension
- The twist beam suspension was validated for kinematics and compliance tests against an ADAMS/Car model of comparable fidelity
- The twist beam suspension was implemented in a full vehicle model and some chassis experiments were run
- Suitability of the beam model in real time vehicle dynamics applications was investigated

In its current implementation, the Euler-Bernoulli beam model provides a simplified method of including component flexibility in mechanical systems in the Dymola environment. Specifically in VDL, effect of component flexibility on overall vehicle performance can be evaluated earlier in the design cycle and parameter sensitivity studies in terms of location and stiffness of beam like components can be rapidly evaluated. This data can be further used to specify cross section dimensions to design components.

### 5.1 Implementation in VDL

When used in a chassis model, no modifications are required to the suspension assembly and the twist beam suspension can easily be used in place of the existing twist beam in VDL and vice versa. Though modifications were done to the state selection attributes in the existing model, the end user is presented with parameters similar to other linkage models in VDL (The bushing and cross beam properties need to be defined in addition to the hardpoints, mass etc).

### 5.2 ADAMS/Car Benchmarking

The suspension model shows good agreement to some specific variables of interest when compared to an ADAMS/Car model of comparable fidelity. Some differences are observed in the toe angle during the kinematics test which can be possibly removed by running a static setup experiment on the suspension assembly first as the toe and camber angle are not defined by a linkage in a semi dependent suspension (Unlike an independent suspension where the length of the tie rod or toe control linkage defines the toe angle). For the kinematics test, the deformations only in the beam are quite close for the case with the same discretization.

### 5.3 Chassis Experiments in VDL

When used in a chassis model, a compliant twist beam suspension can be simulated using a fixed step solver without any issues. Further performance improvement is required to minimize the turnaround time and overruns. Though the vertical inputs in the opposite wheel travel and the roll test cause beam deformation well beyond the linear region of the Euler-Bernoulli beam, two beam elements are sufficient to represent the major characteristics in the double lane change maneuver for the specific vehicle properties. This shows that depending on the application, the beam model can be easily modified in terms of discretization as well as dynamic modes (Dynamic modes were not used in the chassis experiments as part of this thesis).

## 6 Future Work

The potential improvements to the work done as part of thesis can be divided into three specific activities as shown in the following sections:

### 6.1 Beam Model

In terms of the beam model itself, some of the improvements which can be made are:

- Implementation of cross section offset and warping effects and subsequently open cross section geometries.
- Implementation of a beam model suitable for large deformation (e.g. using Timoshenko beam).
- Investigation into asymmetric behavior of the beam.

### 6.2 Twist Beam Suspension Model

Increased model fidelity for the implemented twist beam model can be achieved by:

- Modeling flexible trailing arms
- Running dynamic analysis on the chassis model (using a four poster test rig) and investigating the contribution of the dynamic modes in the FlexBeam.
- Experiments using cross beam location and orientation on full vehicle behavior.
- Investigation of different maneuvers and required discretization of the beam to maintain accuracy.
- Use of two beams with the cross sections offset from each other (one for twist and one for bending) to model shear center offset effects. The multibody modeling and simulation environment Altair MotionView [1] has such a twist beam template.
- Modification of the chassis model to reflect realistic mass and roll stiffness distribution for a compact car.

### 6.3 Additional Applications of the Beam Model

Some additional applications for the current beam model are the following:

- Leaf spring suspension.
- Linkage modeling in independent suspension to optimize compliance behavior.
- Anti roll bar.
- Using a general cross section to model stiffness and inertia distribution in a heavy vehicle chassis.

### 6.4 VDL Usage in Vehicle Dynamics CAE

Additionally, modal superposition combined with the floating frame of reference can be used to implement general modal bodies in VDL, where the modal reduction has been done in an external FE code (Similar to using ABAQUS to generate ADAMS MNF as part of this thesis). The beam model combined with modal bodies will allow VDL to be used as a vehicle dynamics development tool during the entire vehicle development cycle while maintaining the current flexibility offered in terms of scaling of model detail and increased cross domain usage of detailed vehicle models.

# References

- [1] Altair. *MotionView 13.0 Help*. 2016. URL: [http://www.altairhyperworks.com/hwhelp/Altair/hw13.0/help/mv/mv.htm?rear\\_twist\\_beam\\_suspension\\_system\\_svdd.htm](http://www.altairhyperworks.com/hwhelp/Altair/hw13.0/help/mv/mv.htm?rear_twist_beam_suspension_system_svdd.htm).
- [2] Andreasson, Johan and Modelon AB. “The vehicle dynamics library: New concepts and new fields of application”. *8th Modelica Conference, Dresden, Germany*. 2011, pp. 20–22.
- [3] J. Andreasson and M. Gäfvert. “The VehicleDynamics Library-Overview and Applications”. *Proc. 5th International Modelica Conference, Vienna, Austria*. 2006.
- [4] J. Andreasson et al. “Deployment of high-fidelity vehicle models for accurate real-time simulation”. *The First Japanese Modelica Conferences, May 23-24, Tokyo, Japan*. 124. Linköping University Electronic Press. 2016, pp. 78–86.
- [5] J. Andreasson et al. *On Generic Road Vehicle Motion Modelling and Control*. English. 2006.
- [6] T. Best et al. “Design and Operation of a New Vehicle Suspension Kinematics and Compliance Facility”. *SAE Technical Paper*. SAE International, Feb. 1997. DOI: 10.4271/970096. URL: <http://dx.doi.org/10.4271/970096>.
- [7] M. Blundell and D. Harty. *The multibody systems approach to vehicle dynamics*. English. Second edition. Oxford, England; Waltham, Massachusetts; Butterworth-Heinemann, 2015. ISBN: 0080994288.
- [8] K. Chiku et al. “Development of First Order Analysis for Torsion Beam suspension (FOA/TB) corresponding to modular design”. English. Vol. 195. 2013. Chap. 7, pp. 561–569. ISBN: 1876-1100.
- [9] R. R. Craig. Coupling of substructures for dynamic analyses. *AIAA journal* **6.7** (1968), 1313–1319.
- [10] Dassault Systèmes. *ABAQUS 2016 Documentation*. 2016.
- [11] Dassault Systèmes. *Dymola*. 2016. URL: <http://www.3ds.com/products-services/catia/products/dymola>.
- [12] Dassault Systèmes. *Dymola 2016 Documentation*. 2016.
- [13] M. Dempsey, G. Fish, and A. Picarelli. “Using Modelica models for Driver-in-the-loop simulators”. *Furstenfeldbruck, Germany: Proceedings 9th Modelica Conference*. 2012.
- [14] DLR Institute for System Dynamics and Controls. *Flexible Bodies Library*. 2016. URL: <http://www.dlr.de/rmc/rm/en/desktopdefault.aspx/tabid-9281/>.
- [15] H. Elmqvist, S. E. Mattsson, and H. Olsson. “Parallel model execution on many cores”. *Proceedings of the 10th International Modelica Conference*. 2014, pp. 10–12.
- [16] H. Elmqvist et al. Realtime simulation of detailed vehicle and powertrain dynamics. *SAE SP* (2004), 63–76.
- [17] A. Ericsson and A. Kjellander. *Efficient Modeling of a Flexible Beam in Dymola using Coupled Substructures in a Floating Frame of Reference Formulation*. eng. Student Paper. 2015.
- [18] G. Fichera, M. Lacagnina, and F. Petrone. Modelling of Torsion Beam Rear Suspension by Using Multibody Method. *Multibody System Dynamics* **12.4** (), 303–316. ISSN: 1573-272X. DOI: 10.1007/s11044-004-2516-1. URL: <http://dx.doi.org/10.1007/s11044-004-2516-1>.
- [19] Functional Mock-Up Interface Standard. *Functional Mock-Up Interface*. 2016. URL: <https://www.fmi-standard.org/>.
- [20] Heissing, Bernd and Ersoy, Metin and SpringerLink (e-book collection). *Chassis handbook: fundamentals, driving dynamics, components, mechatronics, perspectives*. English. 1st. Wiesbaden: Vieweg + Teubner, 2011. ISBN: 9783834809940; 3834809942;
- [21] D. Lee and C. Yang. An Analytical Approach for Design and Performance Evaluation of Torsion Beam Rear Suspension. *Finite Elem. Anal. Des.* **63** (Jan. 2013), 98–106. ISSN: 0168-874X. DOI: 10.1016/j.finel.2012.09.002. URL: <http://dx.doi.org/10.1016/j.finel.2012.09.002>.
- [22] J. Lee. A structural design of multilevel decomposition and domain mapping. *Journal of Central South University* **21.9** (2014), 3504–3512. ISSN: 2227-5223. DOI: 10.1007/s11771-014-2329-1. URL: <http://dx.doi.org/10.1007/s11771-014-2329-1>.
- [23] J. Lee. A study on the structural characteristics and roll behavior of suspension for the section profile of torsion beam. *Transactions of the Korean Society of Automotive Engineers* **7** (1999), 195–202. URL: [http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JDCGCC\\_1999\\_v7n9\\_195](http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JDCGCC_1999_v7n9_195).
- [24] Modelica and the Modelica Association. *Modelica*. 2016. URL: <https://modelica.org/>.
- [25] Modelica Association. *Modelica Language Specification (version 3.3 Rev 1)*. 2016. URL: <https://www.modelica.org/documents/ModelicaSpec33Revision1.pdf>.

- [26] Modelon. *Vehicle Dynamics Library*. 2016. URL: <http://www.modelon.com/products/modelica-libraries/vehicle-dynamics-library/>.
- [27] Modelon. *Vehicle Dynamics Library Documentation*. 2016.
- [28] MSC Software. *ADAMS flex documentation*. 2016. URL: [https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC10822&cat=2015\\_ADAMS\\_DOCS&actp=LIST](https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC10822&cat=2015_ADAMS_DOCS&actp=LIST).
- [29] MSC Software. *ADAMS MaxFlex*. 2016. URL: <http://www.mscsoftware.com/product/adams-maxflex>.
- [30] MSC Software. *ADAMS/Car documentation*. 2016. URL: <https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC10872&cat=1VM050&actp=LIST>.
- [31] MSC Software. *ADAMS/View documentation*. 2016. URL: <https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC10885&cat=1VM050&actp=LIST>.
- [32] MSC Software. *MSC ADAMS*. 2016. URL: <http://www.mscsoftware.com/product/adams>.
- [33] OptimumG. *Tech tip: Springs and Dampers, Part Two*. 2016. URL: [http://www.optimumg.com/docs/Springs&Dampers\\_Tech\\_Tip\\_2.pdf](http://www.optimumg.com/docs/Springs&Dampers_Tech_Tip_2.pdf).
- [34] T. L. Satchell. *DESIGN OF TRAILING TWIST AXLES*. Tech. rep. 1981.
- [35] A. A. Shabana. *Dynamics of multibody systems*. English. 3rd. New York;Cambridge, U.K; Cambridge University Press, 2005. ISBN: 9780521850117.
- [36] A. Toso and A. Moroni. *Professional Driving Simulator to Design First-Time-Right Race Cars*. English. Tech. rep. 1. 2014.
- [37] L. S. Vasconcelos, M. E. Silveira, and A. L. Christoforo. *Evaluation of the Torsion Beam of a Twist-Beam Suspension by Numerical Simulation*. Tech. rep. SAE Technical Paper, 2012.

# Appendices

## A ABAQUS Input Decks

### A.1 ABAQUS Input deck for unit test beam MNF generation(\*.inp format)

```
*Heading
** Job name: adams_unit_test_beam Model name: Model-1
** Generated by: Abaqus/CAE 2016
** Preprint, echo=NO, model=NO, history=NO, contact=NO
** _____
**
** PART INSTANCE: Part-1-1
**
**
**Node
      1,      -500.,      0.,      0.
      2,      -400.,      0.,      0.
      3,      -300.,      0.,      0.
      4,      -200.,      0.,      0.
      5,      -100.,      0.,      0.
      6,         0.,      0.,      0.
      7,       100.,      0.,      0.
      8,       200.,      0.,      0.
      9,       300.,      0.,      0.
     10,       400.,      0.,      0.
     11,       500.,      0.,      0.
*Element, type=B33
  1,  1,  2
  2,  2,  3
  3,  3,  4
  4,  4,  5
  5,  5,  6
  6,  6,  7
  7,  7,  8
  8,  8,  9
  9,  9, 10
10, 10, 11
*Nset, nset=Part-1-1-Set-1, generate
  1, 11, 1
*Elset, elset=Part-1-1-Set-1, generate
  1, 10, 1
*Nset, nset=Part-1-1-Set-2, generate
  1, 11, 1
*Elset, elset=Part-1-1-Set-2, generate
  1, 10, 1
** Section: Section-1 Profile: Profile-1
*Beam General Section, elset=Part-1-1-Set-1, poisson = 0.3, lumped=NO,
density=8050e-09, section=GENERAL
927., 687000., 0., 665000., 6212.
0.,0.,-1.
210000., 80769.2
*System
*Nset, nset=Set-1
  1, 11
** _____
```

```

**
*STEP
*FREQUENCY,EIGENSOLVER=LANCZOS
20,
*BOUNDARY
Set -1, 1,6
*END STEP
**
**
**
**
**
SUBSTRUCTURE GENERATION
**
*STEP
*SUBSTRUCTURE GENERATE, TYPE=Z1, RECOVERY MATRIX=YES,
MASS MATRIX=YES, OVERWRITE
*FLEXIBLE BODY, TYPE=ADAMS
*RETAINED NODAL DOFS
Set -1, 1,6
*SELECT EIGENMODES, generate
1,20
*END STEP

```

## A.2 ABAQUS Input deck for twist beam MNF generation(\*.inp format)

```

*Heading
** Job name: ADAMS_twist_beam Model name: ADAMS_twist_beam
** Generated by: Abaqus/CAE 2016
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PART INSTANCE: Part-1-1
**
*Node
1, -240., 700., 0.
2, -240., 560., 0.
3, -240., 420., 0.
4, -240., 280., 0.
5, -240., 140., 0.
6, -240., 0., 0.
7, -240., -140., 0.
8, -240., -280., 0.
9, -240., -420., 0.
10, -240., -560., 0.
11, -240., -700., 0.
*Element, type=B33
1, 1, 2
2, 2, 3
3, 3, 4
4, 4, 5
5, 5, 6
6, 6, 7
7, 7, 8
8, 8, 9
9, 9, 10
10, 10, 11

```

```

*Nset, nset=Beam, generate
  1, 11, 1
*Elset, elset=Beam, generate
  1, 10, 1
** Section: Section-1 Profile: Profile-1
*Beam General Section, elset=Beam, poisson = 0.3, lumped=NO, density=8.05e-06,
  section=GENERAL
927., 687000., 0., 665000., 6212.
0., 0., -1.
210000., 80769.2
*System
*Nset, nset=RETNODES
  1, 11
**
**
*STEP
*FREQUENCY, EIGENSOLVER=LANCZOS
20,
*BOUNDARY
RETNODES, 1, 6
*END STEP
**
**
**
SUBSTRUCTURE GENERATION
**
*STEP
*SUBSTRUCTURE GENERATE, TYPE=Z1, RECOVERY MATRIX=YES,
  MASS MATRIX=YES, OVERWRITE
*FLEXIBLE BODY, TYPE=ADAMS
*RETAINED NODAL DOFS
RETNODES, 1, 6
*SELECT EIGENMODES, generate
1, 20
*END STEP

```

## B ABAQUS Output

### B.1 Partial ABAQUS output for substructure generation(\*.dat format)

| S U B S T R U C T U R E   E I G E N V A L U E   O U T P U T |             |            |               |                  |
|-------------------------------------------------------------|-------------|------------|---------------|------------------|
| MODE NO                                                     | EIGENVALUE  | FREQUENCY  |               | GENERALIZED MASS |
|                                                             |             | (RAD/TIME) | (CYCLES/TIME) |                  |
| 1                                                           | -5.6128E-11 | 0.00000    | 0.00000       | 1.0000           |
| 2                                                           | -2.2383E-11 | 0.00000    | 0.00000       | 1.0000           |
| 3                                                           | 2.9195E-13  | 5.4032E-07 | 8.5995E-08    | 1.0000           |
| 4                                                           | 4.5519E-11  | 6.7468E-06 | 1.0738E-06    | 1.0000           |
| 5                                                           | 5.2888E-11  | 7.2724E-06 | 1.1574E-06    | 1.0000           |
| 6                                                           | 5.3095E-11  | 7.2866E-06 | 1.1597E-06    | 1.0000           |

|    |            |        |        |        |
|----|------------|--------|--------|--------|
| 7  | 234.05     | 15.299 | 2.4349 | 1.0000 |
| 8  | 959.49     | 30.976 | 4.9299 | 1.0000 |
| 9  | 2248.0     | 47.413 | 7.5460 | 1.0000 |
| 10 | 2439.0     | 49.386 | 7.8600 | 1.0000 |
| 11 | 2519.7     | 50.196 | 7.9890 | 1.0000 |
| 12 | 4223.2     | 64.986 | 10.343 | 1.0000 |
| 13 | 7056.2     | 84.001 | 13.369 | 1.0000 |
| 14 | 10925.     | 104.52 | 16.635 | 1.0000 |
| 15 | 15867.     | 125.96 | 20.048 | 1.0000 |
| 16 | 18569.     | 136.27 | 21.688 | 1.0000 |
| 17 | 19183.     | 138.50 | 22.043 | 1.0000 |
| 18 | 21436.     | 146.41 | 23.302 | 1.0000 |
| 19 | 26249.     | 162.02 | 25.786 | 1.0000 |
| 20 | 28225.     | 168.00 | 26.738 | 1.0000 |
| 21 | 71493.     | 267.38 | 42.555 | 1.0000 |
| 22 | 73858.     | 271.77 | 43.253 | 1.0000 |
| 23 | 1.5972E+05 | 399.64 | 63.605 | 1.0000 |
| 24 | 1.9684E+05 | 443.67 | 70.612 | 1.0000 |
| 25 | 2.0335E+05 | 450.95 | 71.771 | 1.0000 |
| 26 | 4.4101E+05 | 664.09 | 105.69 | 1.0000 |
| 27 | 4.5560E+05 | 674.98 | 107.43 | 1.0000 |
| 28 | 6.9345E+05 | 832.74 | 132.53 | 1.0000 |
| 29 | 3.5343E+06 | 1880.0 | 299.21 | 1.0000 |
| 30 | 3.6512E+06 | 1910.8 | 304.12 | 1.0000 |
| 31 | 6.7431E+06 | 2596.7 | 413.28 | 1.0000 |
| 32 | 6.9662E+06 | 2639.3 | 420.07 | 1.0000 |

## C ADAMS/Car files

### C.1 ADAMS/Car assembly file(TeimOrbit format)

```

$-----
MDLHEADER
[MDLHEADER]
FILE_TYPE      =  'asy'
FILE_VERSION   =  2.0
FILE_FORMAT    =  'ASCII'
HEADER_SIZE    =  9
(COMMENTS)
{comment_string}
'Adams/Car suspension assembly'
$-----
ASSEMBLY_HEADER
[ASSEMBLY_HEADER]
ASSEMBLY_CLASS =  'suspension'
TIMESTAMP      =  '2016/03/21,14:02:02'
HEADER_SIZE    =  5
$-----
PLUGINS
[PLUGINS]
PLUGIN_LIST    =  'acar'
HEADER_SIZE    =  4
$-----
UNITS
[UNITS]

```

LENGTH = 'mm'  
 FORCE = 'newton'  
 ANGLE = 'deg'  
 MASS = 'kg'  
 TIME = 'sec'

\$

#### SUBSYSTEM

[SUBSYSTEM]

\$ Subsystem information:

\$ Major Role : suspension

\$ Minor Role : rear

\$ Template : \_twist\_beam\_B33\_beam

USAGE = '<Twist\_beam\_B33>/subsystems.tbl/twist\_beam\_B33.sub'

\$

#### TESTRIG

[TESTRIG]

USAGE = '\_MDL.SUSPENSION.TESTRIG'

\$

#### HARDPOINT

[HARDPOINT]

| {hardpoint_name             |  | symmetry |  | x_value | y_value | z_value |
|-----------------------------|--|----------|--|---------|---------|---------|
| }                           |  |          |  |         |         |         |
| 'global_part_reference      |  | 'single  |  | 0.0     | 0.0     |         |
| 0.0                         |  |          |  |         |         |         |
| 'steering_input_rotation    |  | 'single  |  | 200.0   | 200.0   |         |
| 0.0                         |  |          |  |         |         |         |
| 'steering_input_slider      |  | 'single  |  | 200.0   | 0.0     |         |
| 0.0                         |  |          |  |         |         |         |
| 'steering_input_translation |  | 'single  |  | 200.0   | -200.0  |         |
| 0.0                         |  |          |  |         |         |         |

\$

#### PARAMETER

[PARAMETER]

| {parameter_name             |      | symmetry |  | type      | value}        |
|-----------------------------|------|----------|--|-----------|---------------|
| 'compliance_matrix_flag     |      | 'single  |  | 'integer' | 1             |
| 'dynamic_force_flag         |      | 'single  |  | 'integer' | 0             |
| 'dynamic_motion_flag        |      | 'single  |  | 'integer' | 0             |
| 'ISO_mode_flag              |      | 'single  |  | 'integer' | 0             |
| 'axle_ratio                 |      | 'single  |  | 'real'    | 1.0           |
| 'brake_ratio                |      | 'single  |  | 'real'    | 0.55          |
| 'cg_height                  |      | 'single  |  | 'real'    | 300.0         |
| 'drive_ratio                |      | 'single  |  | 'real'    | 0.5           |
| 'max_jack_force             |      | 'single  |  | 'real'    | 40000.0       |
| 'min_jack_force             |      | 'single  |  | 'real'    | -20000.0      |
| 'testrig_tire_property_file |      | 'single  |  | 'string'  | 'RIGID.WHEEL' |
| 'testrig_wheel_radius       |      | 'single  |  | 'real'    | 293.2         |
| 'Used by suspension testrig | tire |          |  |           |               |
| 'tire_stiffness             |      | 'single  |  | 'real'    | 200.0         |
| 'total_sprung_mass          |      | 'single  |  | 'real'    | 1200.0        |
| 'wheelbase                  |      | 'single  |  | 'real'    | 2000.0        |
| 'wheel_mass                 |      | 'single  |  | 'real'    | 1.0           |

\$

#### SOLVER.SETTINGS

[SOLVER.SETTINGS]

```

INTEGRATOR      = 'gstiff'
CORRECTOR       = 'modified'
FORMULATION     = 'I3'

```

## C.2 ADAMS/Car subsystem file(TeimOrbit format)

```

$-----
MDLHEADER
[MDLHEADER]
FILE_TYPE      = 'sub'
FILE_VERSION   = 7.0
FILE_FORMAT    = 'ASCII'
$-----

SUBSYSTEMHEADER
[SUBSYSTEMHEADER]
TEMPLATE_NAME  = 'mdids://Twist_beam_B33/templates.tbl/_twist_beam_B33_beam.
tpl'
MAJOR_ROLE     = 'suspension'
MINOR_ROLE     = 'rear'
TIMESTAMP     = '2016/03/22,13:31:06'
$-----

UNITS
[UNITS]
LENGTH        = 'mm'
FORCE         = 'newton'
ANGLE         = 'deg'
MASS          = 'kg'
TIME          = 'sec'
$-----

HARDPOINT
[HARDPOINT]
{hardpoint_name      symmetry      x_value      y_value      z_value
}
'body_ref            ' 'single'      0.0          0.0
150.0
'flex_body_ref       ' 'single'      -400.0        0.0
200.0
'beam_left           ' 'left/right'  -240.0        -700.0
500.0
'damper_lower        ' 'left/right'  1.6556718178 -744.1116381653
542.090108041
'damper_upper        ' 'left/right'  -28.3443281822 -719.1116381653
942.090108041
'drive_shaft_inr     ' 'left/right'  -50.0         -200.0
550.0
'spring_lower        ' 'left/right'  -117.174489175 -745.6087228331
542.2725287254
'spring_upper        ' 'left/right'  -117.174489175 -745.6087228331
742.2725287254
'trailing_arm_body   ' 'left/right'  -400.0        -750.0
500.0
'wheel_center        ' 'left/right'  0.0          -850.0
500.0
$-----

PART_ASSEMBLY
[PART_ASSEMBLY]

```

```

USAGE                      = 'drive_shaft '
SYMMETRY                   = 'left/right '
MODE                       = 'rigid '
$ Rigid body data:
  MASS                     = 4.2174529406
$ Part location is dependent.
$   X,Y,Z location = -50.0, -200.0, 550.0
$ Part orientation is dependent.
$   ZP vector = -0.0776524845, 0.9939518013, 0.0776524845
$   XP vector = -0.7028300589, -0.1098171967, 0.7028300589
CMLOCATIONFROMPARTX = -2.5374265291E-15
CMLOCATIONFROMPARTY = -2.5374265291E-15
CMLOCATIONFROMPARTZ = -365.6817014651
IXX                     = 1.6598906562E+05
IYY                     = 1.6598906562E+05
IZZ                     = 692.8258536527
IXY                     = 0.0
IZX                     = 0.0
IYZ                     = 0.0
$
  LINK.GEOMETRY
[LINK.GEOMETRY]
  USAGE                  = 'drive_shaft '
  PART                   = 'drive_shaft '
  SYMMETRY               = 'left/right '
  RADIUS                 = 15.0
$
  PART.ASSEMBLY
[PART.ASSEMBLY]
  USAGE                  = 'lower_strut '
  SYMMETRY               = 'left/right '
  MODE                   = 'rigid '
$ Rigid body data:
  MASS                     = 3.032811081
$ Part location is dependent.
$   X,Y,Z location = -8.5443281822, -735.6116381653, 678.090108041
$ Part orientation is dependent.
$   ZP vector = -0.074645113, 0.0622042608, 0.9952681734
$   XP vector = 0.9971993099, -5.5511151231E-16, 0.0747899482
CMLOCATIONFROMPARTX = 0.0
CMLOCATIONFROMPARTY = 0.0
CMLOCATIONFROMPARTZ = 0.0
IXX                     = 10382.070533
IYY                     = 10382.070533
IZZ                     = 947.7534628024
IXY                     = 0.0
IZX                     = 0.0
IYZ                     = 0.0
$
  PART.ASSEMBLY
[PART.ASSEMBLY]
  USAGE                  = 'spindle '
  SYMMETRY               = 'left/right '
  MODE                   = 'rigid '
$ Rigid body data:
  MASS                     = 1.1028403931

```

```

$ Part location is dependent.
$   X,Y,Z location = 0.0, -850.0, 500.0
$ Part orientation is dependent.
$   ZP vector = 0.0, -1.0, 0.0
$   XP vector = 1.0, 0.0, 0.0
CMLOCATIONFROMPARTX = 0.0
CMLOCATIONFROMPARTY = 0.0
CMLOCATIONFROMPARTZ = 15.0
IXX                = 477.8975036678
IYY                = 477.8975036678
IZZ                = 496.2781768857
IXY                = 0.0
IZX                = 0.0
IYZ                = 0.0
$-----
$ PART ASSEMBLY
[PART ASSEMBLY]
  USAGE              = 'trailing_arm_2 '
  SYMMETRY           = 'left/right '
  MODE               = 'rigid '
$ Rigid body data:
  MASS               = 0.6735054266
$ Part location is dependent.
$   X,Y,Z location = -120.0, -775.0, 500.0
$ Part orientation is dependent.
$   ZP vector = 0.847998304, -0.52999894, 0.0
$   XP vector = -9.7359225814E-17, -1.557747613E-16, -1.0
CMLOCATIONFROMPARTX = -2.40589927E-16
CMLOCATIONFROMPARTY = 5.4978587175E-14
CMLOCATIONFROMPARTZ = -2.7755575616E-14
IXX                = 4512.0011280267
IYY                = 4512.0011280267
IZZ                = 32.7048112716
IXY                = 4.1305610888E-13
IZX                = 5.1947511735E-13
IYZ                = 2.9396923216E-13
$-----
$ LINK GEOMETRY
[LINK GEOMETRY]
  USAGE              = 'trailing_arm_2 '
  PART               = 'trailing_arm_2 '
  SYMMETRY           = 'left/right '
  RADIUS             = 10.0
$-----
$ PART ASSEMBLY
[PART ASSEMBLY]
  USAGE              = 'trailing_arm_1 '
  SYMMETRY           = 'left/right '
  MODE               = 'rigid '
$ Rigid body data:
  MASS               = 0.398912827
$ Part location is dependent.
$   X,Y,Z location = -320.0, -725.0, 500.0
$ Part orientation is dependent.
$   ZP vector = 0.954479978, 0.2982749931, 0.0
$   XP vector = 5.4792227341E-17, -1.7533512749E-16, -1.0

```

```

CMLOCATIONFROMPARTX = -4.2494077655E-16
CMLOCATIONFROMPARTY = -2.1350390285E-14
CMLOCATIONFROMPARTZ = 2.7755575616E-14
IXX                = 943.8062919265
IYY                = 943.8062919265
IZZ                = 19.3708442498
IXY                = 3.7638450323E-14
IZX                = 2.8312043514E-14
IYZ                = 3.3001784003E-16
$
  LINK_GEOMETRY
[LINK_GEOMETRY]
  USAGE            = 'traling_arm_1'
  PART             = 'traling_arm_1'
  SYMMETRY         = 'left/right'
  RADIUS           = 10.0
$
  PART_ASSEMBLY
[PART_ASSEMBLY]
  USAGE            = 'tripot'
  SYMMETRY         = 'left/right'
  MODE             = 'rigid'
$ Rigid body data:
  MASS             = 1.9851127075
$ Part location is dependent.
$ X,Y,Z location = -50.0, -200.0, 550.0
$ Part orientation is dependent.
$ ZP vector = 1.2246467991E-16, 1.0, 0.0
$ XP vector = 0.7071067812, -8.6595605624E-17, -0.7071067812
  CMLOCATIONFROMPARTX = 0.0
  CMLOCATIONFROMPARTY = 0.0
  CMLOCATIONFROMPARTZ = 13.8888888889
  IXX                = 1101.8600905078
  IYY                = 1101.8600905078
  IZZ                = 813.8962100926
  IXY                = 0.0
  IZX                = 0.0
  IYZ                = 0.0
$
  PART_ASSEMBLY
[PART_ASSEMBLY]
  USAGE            = 'upper_strut'
  SYMMETRY         = 'left/right'
  MODE             = 'rigid'
$ Rigid body data:
  MASS             = 4.3672479566
$ Part location is dependent.
$ X,Y,Z location = -18.4443281822, -727.3616381653, 810.090108041
$ Part orientation is dependent.
$ ZP vector = -0.074645113, 0.0622042608, 0.9952681734
$ XP vector = 0.9971993099, -4.9960036108E-16, 0.0747899482
  CMLOCATIONFROMPARTX = 0.0
  CMLOCATIONFROMPARTY = 0.0
  CMLOCATIONFROMPARTZ = 0.0
  IXX                = 15250.429864
  IYY                = 15250.429864

```

```

IZZ          = 1965.2615804675
IXY          = 0.0
IZX          = 0.0
IYZ          = 0.0

```

\$

# FLEXIBLE\_BODY

[FLEXIBLE\_BODY]

```

USAGE          = 'B33_twist_beam'
SYMMETRY       = 'single'
FLEX_BODY_LOC_X = 0.0
FLEX_BODY_LOC_Y = 0.0
FLEX_BODY_LOC_Z = 500.0

```

\$ Flexible body orientation is dependent.

\$ ZP vector = 0.0, 0.0, 1.0

\$ XP vector = 1.0, 0.0, 0.0

```

MODAL_NEUTRAL_FILE = '<Twist_beam_B33>/flex_bodys.tbl/ADAMS_twist_beam.mnf'
DAMPING_RATIO      = 'default'
MASS_INVARIANTS    = 'user_defined'
INVARIANT_1        = 'yes'
INVARIANT_2        = 'yes'
INVARIANT_3        = 'yes'
INVARIANT_4        = 'yes'
INVARIANT_5        = 'no'
INVARIANT_6        = 'yes'
INVARIANT_7        = 'yes'
INVARIANT_8        = 'yes'
INVARIANT_9        = 'no'

```

(MODESTATUS)

| {mode_number | mode_status | info:frequency} |
|--------------|-------------|-----------------|
| 1            | 'disabled'  | -0.00004        |
| 2            | 'disabled'  | -0.00002        |
| 3            | 'disabled'  | 0.00000         |
| 4            | 'disabled'  | 0.00003         |
| 5            | 'disabled'  | 0.00004         |
| 6            | 'disabled'  | 0.00004         |
| 7            | 'enabled'   | 76.99787        |
| 8            | 'enabled'   | 155.89789       |
| 9            | 'enabled'   | 238.62596       |
| 10           | 'enabled'   | 248.55553       |
| 11           | 'enabled'   | 252.63352       |
| 12           | 'enabled'   | 327.06959       |
| 13           | 'enabled'   | 422.77087       |
| 14           | 'enabled'   | 526.04521       |
| 15           | 'enabled'   | 633.96541       |
| 16           | 'enabled'   | 685.82229       |
| 17           | 'enabled'   | 697.07441       |
| 18           | 'enabled'   | 736.86612       |
| 19           | 'enabled'   | 815.41544       |
| 20           | 'enabled'   | 845.54173       |
| 21           | 'enabled'   | 1345.71179      |
| 22           | 'enabled'   | 1367.79057      |
| 23           | 'enabled'   | 2011.38138      |
| 24           | 'enabled'   | 2232.94832      |
| 25           | 'enabled'   | 2269.58378      |
| 26           | 'enabled'   | 3342.29397      |
| 27           | 'enabled'   | 3397.13020      |

|    |            |             |
|----|------------|-------------|
| 28 | 'enabled ' | 4191.10688  |
| 29 | 'enabled ' | 9461.76983  |
| 30 | 'enabled ' | 9617.00684  |
| 31 | 'enabled ' | 13069.20438 |
| 32 | 'enabled ' | 13283.62771 |

---

\$

BUSHING\_ASSEMBLY

[BUSHING\_ASSEMBLY]

DEFINITION = '.ACAR.attachments.ac\_bushing'

USAGE = 'beam\_to\_body'

SYMMETRY = 'left/right'

\$ Bushing orientation is dependent.

\$ ZP vector = 1.2246467991E-16, 1.0, 0.0

\$ XP vector = 1.0, -1.2246467991E-16, 1.2246467991E-16

T\_PRELOAD\_X = 0.0

T\_PRELOAD\_Y = 0.0

T\_PRELOAD\_Z = 0.0

R\_PRELOAD\_X = 0.0

R\_PRELOAD\_Y = 0.0

R\_PRELOAD\_Z = 0.0

T\_OFFSET\_X = 0.0

T\_OFFSET\_Y = 0.0

T\_OFFSET\_Z = 0.0

R\_OFFSET\_X = 0.0

R\_OFFSET\_Y = 0.0

R\_OFFSET\_Z = 0.0

FX\_SCALING\_FACTOR = 1.0

FY\_SCALING\_FACTOR = 1.0

FZ\_SCALING\_FACTOR = 1.0

TX\_SCALING\_FACTOR = 1.0

TY\_SCALING\_FACTOR = 1.0

TZ\_SCALING\_FACTOR = 1.0

TX\_DAMPING\_FORCE\_SCALE = 1.0

TY\_DAMPING\_FORCE\_SCALE = 1.0

TZ\_DAMPING\_FORCE\_SCALE = 1.0

RX\_DAMPING\_FORCE\_SCALE = 1.0

RY\_DAMPING\_FORCE\_SCALE = 1.0

RZ\_DAMPING\_FORCE\_SCALE = 1.0

PROPERTY\_FILE = '<Twist\_beam\_B33>/bushings.tbl/mdi\_0001.bus'

---

\$

NSPRING\_ASSEMBLY

[NSPRING\_ASSEMBLY]

DEFINITION = '.ACAR.forces.ac\_spring'

USAGE = 'ride\_spring'

SYMMETRY = 'left/right'

PROPERTY\_FILE = '<Twist\_beam\_B33>/springs.tbl/mdi\_0001.spr'

VALUE\_TYPE = 'installed\_length'

USER\_VALUE = 130.0

---

\$

DAMPER\_ASSEMBLY

[DAMPER\_ASSEMBLY]

DEFINITION = '.ACAR.forces.ac\_damper'

USAGE = 'ride\_damper'

SYMMETRY = 'left/right'

PROPERTY\_FILE = '<Twist\_beam\_B33>/dampers.tbl/mdi\_0001.dpr'

---

\$

```

    BUMPSTOP_ASSEMBLY
[BUMPSTOP_ASSEMBLY]
DEFINITION      =  '.ACAR.forces.ac_bumpstop'
USAGE           =  'jounce_bumper'
SYMMETRY        =  'left/right'
PROPERTY_FILE   =  '<Twist_beam_B33>/bumpstops.tbl/mdi_0001.bum'
DISTANCE_TYPE   =  'clearance'
USER_DISTANCE   =  90.0

```

\$

```

    PARAMETER
[PARAMETER]
{parameter_name      symmetry      type      value}
'driveline_active    ' 'single'      'integer'  0
'kinematic_flag      ' 'single'      'integer'  0
'camber_angle        ' 'left/right'  'real'     0.0
'drive_shaft_offset  ' 'left/right'  'real'     10.0
'toe_angle           ' 'left/right'  'real'     0.0

```

## D Dymola translation and simulation logs

### D.1 Example translation log from a VDL suspension experiment

Translation of FlexBeamV1.Multibody.Internal.Concepts.VDLApplications.Chassis.Suspensions.Experiments.SuspensionInRig:

Note: Default ISO Frontward–Leftward–Upward (FLU) reference is used.

Note: "ground" is not defined, drag any VehicleDynamics.Grounds model to the top level of your experiment.

Note: No "aggregateMass" component is defined. A default AggregateMass component will be used.

The DAE has 23684 scalar unknowns and 23684 scalar equations.

Statistics

Original Model

Number of components: 1480

Variables: 11260

Constants: 18 (18 scalars)

Parameters: 4081 (9629 scalars)

Unknowns: 7161 (23684 scalars)

Differentiated variables: 171 scalars

Equations: 5402

Nontrivial: 3796

Translated Model

Constants: 22887 scalars

Free parameters: 4 scalars

Parameter depending: 1772 scalars

Continuous time states: 36 scalars

Time-varying variables: 2583 scalars

Alias variables: 6085 scalars

Number of mixed real/discrete systems of equations: 0

Sizes of linear systems of equations: {3, 3, 3, 3, 3, 3, 3, 210, 6, 6}  
 Sizes after manipulation of the linear systems: {0, 0, 0, 0, 0, 0, 0, 30, 0, 0}  
 Sizes of nonlinear systems of equations: { }  
 Sizes after manipulation of the nonlinear systems: { }  
 Number of numerical Jacobians: 0

Initialization problem

Sizes of linear systems of equations: {3}  
 Sizes after manipulation of the linear systems: {0}

Selected continuous time states

Statically selected continuous time states

suspension.linkage.flexBeam.flexBeamBase[1].q\_f[1]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_f[2]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_f[3]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_f[4]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_f[5]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_f[6]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[1]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[2]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[3]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[4]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[5]  
 suspension.linkage.flexBeam.flexBeamBase[1].q\_fdot[6]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[1]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[2]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[3]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[4]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[5]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_f[6]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[1]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[2]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[3]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[4]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[5]  
 suspension.linkage.flexBeam2.flexBeamBase[1].q\_fdot[6]  
 suspension.linkage.rightBushing1.motion.phi1  
 suspension.linkage.rightBushing1.motion.phi2  
 suspension.linkage.rightBushing1.motion.phi3  
 suspension.linkage.rightBushing1.motion.phi\_d1  
 suspension.linkage.rightBushing1.motion.phi\_d2  
 suspension.linkage.rightBushing1.motion.phi\_d3  
 suspension.linkage.rightBushing1.motion.r\_rel\_a1  
 suspension.linkage.rightBushing1.motion.r\_rel\_a2  
 suspension.linkage.rightBushing1.motion.r\_rel\_a3  
 suspension.linkage.rightBushing1.motion.v\_rel\_a1  
 suspension.linkage.rightBushing1.motion.v\_rel\_a2  
 suspension.linkage.rightBushing1.motion.v\_rel\_a3

Finished

## D.2 Example simulation log from a VDL suspension experiment

Log-file of program ./dymosim  
 (generated: Fri May 13 14:58:37 2016)

```

dymosim started
... "dsin.txt" loading (dymosim input file)
DataAccess.TO, Copyright 2014–2016, Modelon AB. All rights reserved.
DataAccess.TBL, Copyright 2013–2016, Modelon AB. All rights reserved.
... "SuspensionInRig.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 10
  CPU-time for integration      : 0.33 seconds
  CPU-time for one GRID interval: 0.66 milli-seconds
  Number of result points      : 501
  Number of GRID points       : 501
  Number of (successful) steps : 627
  Number of F-evaluations      : 3072
  Number of Jacobian-evaluations: 49
  Number of (model) time events : 1
  Number of (U) time events    : 0
  Number of state events      : 0
  Number of step events       : 0
  Minimum integration stepsize : 2.14e-008
  Maximum integration stepsize : 2.07
  Maximum integration order    : 5
Calling terminal section
... "dsfinal.txt" creating (final states)

```

### D.3 Partial translation log from a VDL suspension experiment running on multiple cores

```

Parallelization

For OutputSection

For DynamicsSection

Estimated operation count for sequential execution: 193162

Estimated operation count for longest path using 8 cores: 189852

Estimated operation count for critical path: 163912

Parallel 1:

Unknowns: Total = 669 and Max = 345
Estimated operation count: Total = 6621 and Max = 3311

Section 1:1

Unknowns: 324
Estimated operation count: 3311

Section 1:2

Unknowns: 345

```

Estimated operation count: 3310

Sequential 2:

Unknowns: 1221

Linear systems: 1

Estimated operation count: 186541