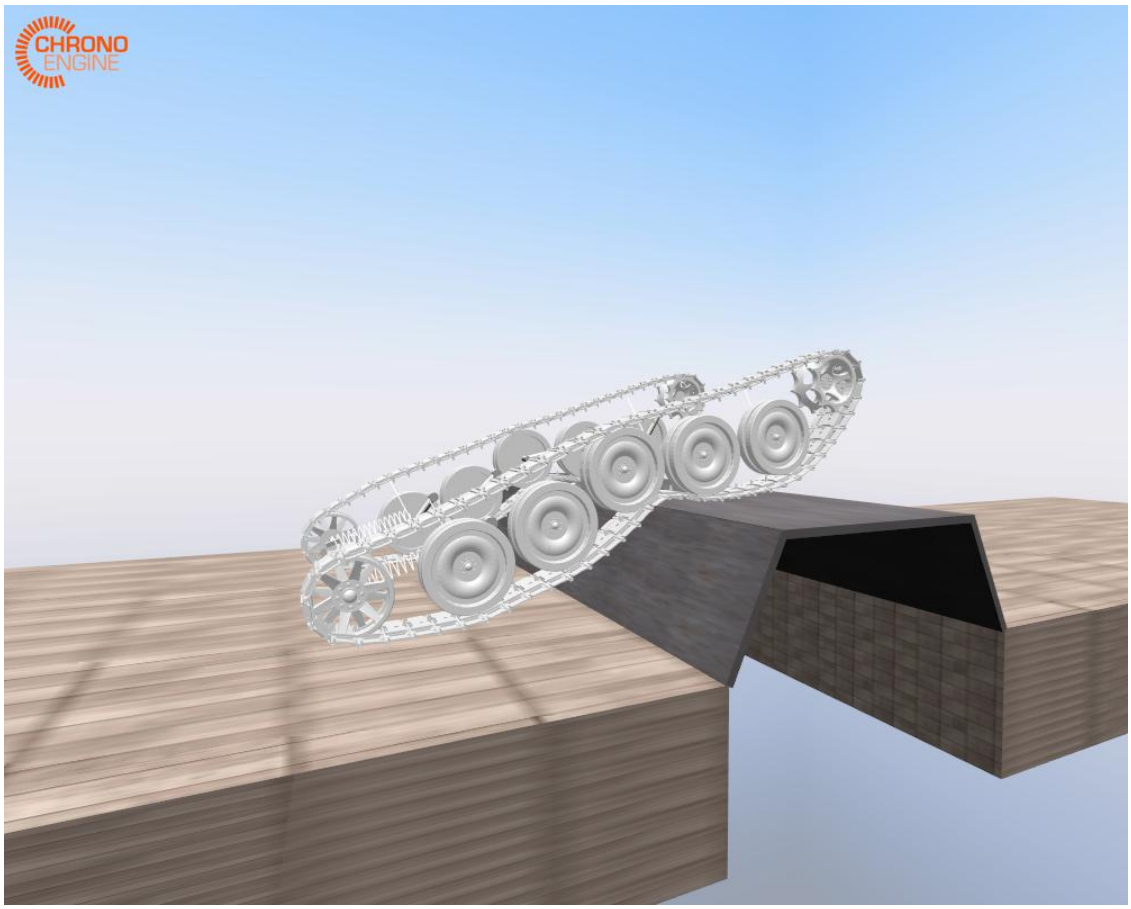

Influence of Track-Assembly Geometry on Mobility Over Rigid Terrain

Master's thesis in Complex Adaptive Systems

ALEXANDER KÖRNER

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

Influence of Track-Assembly Geometry on Mobility Over Rigid Terrain

ALEXANDER KÖRNER

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Division name
Name of research group (if applicable)
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Influence of Track-Assembly Geometry on Mobility Over Rigid Terrain
ALEXANDER KÖRNER

Supervisor: Mathilda Karlsson Hagnell, FOI
Examiner: Adam Andersson, Department of Mathematical Sciences

Master's Thesis 2024
Department of Mathematical Sciences
Division of Division name
Name of research group (if applicable)
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A visual rendering of a simulation of a tracked vehicle traversing a trapezoidal obstacle.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Influence of Track-Assembly Geometry on Mobility Over Rigid Terrain
ALEXANDER KÖRNER
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

This thesis considers the modeling and simulation of tracked vehicles. Specifically it aims to investigate how the geometry of the track-assembly affects its mobility over rigid obstacles. This is done with the purpose of aiding in the Swedish Defence Research Agency's (FOI) goal of developing parametrized virtual models of tracked vehicles for researching mobility.

The vehicle is modeled as a multibody dynamics simulation consisting of rigid bodies. This entails solving index-3 differential algebraic equations (DAE) with high precision. The validation of the vehicle model and simulation included the comparison of two methods of solving the DAE in question, both of which require the solving of a saddle point problem. Thus for each method two iterative saddle point problem solvers have been compared as well. After choosing a numerical method the mobility of the model is evaluated on rigid obstacles with varying track-assembly geometry.

The evaluation shows that the track-assembly geometry significantly impacts the models ability to traverse vertical step obstacles and half round obstacles. The results suggest that this applies to a general real tracked vehicle, however a more thorough model validation and investigation into the numerical methods is required to draw any quantitative conclusions.

Keywords: Project Chrono, Tracked vehicle, Track assembly, vehicle dynamics, vehicle simulation, multibody simulation, vehicle modeling, rigid obstacles.

Contents

1
Introduction

Tracked vehicles are found in a wide variety of scenes, from agricultural vehicles traversing soaking rice paddies to snowmobiles dashing over frozen lakes and snow covered hills. Wherever the terrain is soft or muddy, chances are a tracked vehicle can be found close by. But the perhaps most striking examples of tracked vehicles can be found in the arsenal of the world's military forces, where they are used for everything from carrying portable bridges to penetrating the front lines of the enemy. Regardless of how a tracked vehicle is used, the main purpose of the common feature between all tracked vehicles, the tracks, is to provide mobility over terrain where wheels would fail.

The mobility of a military tracked vehicle has different meaning depending on the type of operation the vehicle is expected to perform. In the widest sense there is the strategic mobility, which alludes to the ability to transport the vehicle to the operational zone. The level below strategic would be the operational mobility which entails the capacity of the vehicle to move long distances under its own power, limited by for example fuel capacity, top speed and ride quality. Lastly there is battlefield mobility, characterized by the ability to traverse all kinds of rough terrain and obstacles at as high of a speed as possible to minimize the exposure to the enemy. Battlefield mobility will be the focus of this thesis. Specifically how the geometry of the track-assembly of a tracked armored military vehicle affects its mobility. [1]

1.1 Scope of Work

This thesis aims to investigate how the geometry of the track-assembly of a vehicle affects its mobility over rigid terrain, by modeling, validating and simulating a virtual vehicle. Specifically, different track-assembly configurations – where the horizontal and vertical position of the idler and sprocket is varied – are evaluated on a rigid vertical step obstacle and multiple rigid half-round obstacles. This is done with the purpose of aiding in the Swedish Defence Research Agency's (FOI) goal of developing parameterized virtual models of tracked vehicles for researching mobility.

To investigate this the physics based simulation engine Chrono [2] will be used to model a parameterized tracked vehicle, inspired by the Armored Personnel Carrier (APC) M113 outlined in "Next Generation NATO Reference Mobility Model (NG-NRMM) Development" [3].

1.2 Outline

The bulk of the work can be summarized as *Modeling*, *Validating* and finally *Evaluating*. In more detail the thesis consist of the following chapters: chapter 2 gives an introduction to Chrono and the theory behind the multibody dynamics it implements, as well as an overview of the relevant parts of a track-assembly and an overview of the NG-NRMM report [3]. Chapter 3 starts by explaining the structure of the vehicle model and how to solve the technicalities of creating a dynamic model in Chrono. It continues by presenting how the vehicle parameters were set to model the M113. Chapter 4 starts by explaining how some of the simulation parameters were set. The chapter continues with a validation of the M113 model, which consists of a comparison between the model and the data found in the NG-NRMM report [3]. In Chapter 5 the model is evaluated on some selected events while varying the sprocket and idler positions. The chapter ends with a discussion of the results. Finally the thesis is concluded in chapter 6.

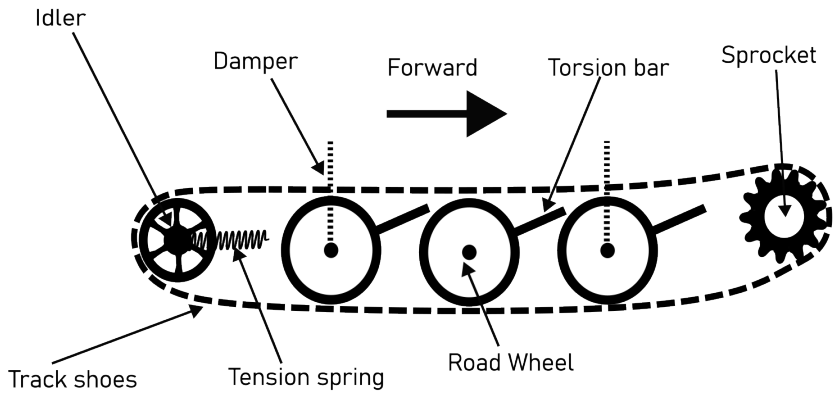

Figure 1.1: A graphical representation of a tracked vehicle where the essential parts in relation to the thesis are annotated

2

Background and Theory

2.1 Track-Assembly of Tracked Vehicles

There are many different ways to construct the track-assembly of a tracked vehicle, figure 1.1 display the layout relevant for this thesis. This assembly places the driving sprocket in the front and the track tensioning idler in the rear. There are five road wheels carrying the load of the vehicle, all are fitted with torsion bar springs and the front and rear are fitted with linear dampers. The actual track consists of individual track shoes connected by single revolute pins, making it a single pin track assembly.

2.2 Chrono

Chrono is an open source multiphysics simulation engine currently being developed at the University of Wisconsin-Madison, USA [2]. It specializes on large complex dynamical systems where the interaction of bodies can be described with frictional contact [4]. There is however also support for fluid dynamics, granular simulations and more. The Chrono library is written in C++ and as a user one can use its API with either C++ or a python interface. There are optional modules available for Chrono which adds specialized functionality, for example the module `Chrono::Vehicle` designed to build parametrized modular models and simulations of wheeled and tracked vehicles. It is this physics based vehicle simulation capability that has attracted attention of – among others – the US army which has granted the project financial aid to continue research at the university [5].

In alignment with the goal to make Chrono a reliable tool for predicting the time evolution of complex systems, the team has performed validation studies where the core simulation elements have been compared with the corresponding elements used in MSC ADAMS [6]. Most of the performed simulations did not have closed form solutions, hence the use of MSC ADAMS. Validation of the `chrono::Vehicle` to MSC ADAMS/CAR has also been performed [7], where the two car models were "... in relatively close agreement" according to the authors. Additional validation studies have been performed with comparisons to experimental data where Chrono has proved predictive [8] [9].

2.3 Next Generation NATO Reference Mobility Model Development

The Next Generation NATO Reference Mobility Model Development is a technical report, henceforth called NG-NRMM, from the NATO Science and Technology Organisation team AVT-248 that summarize the effort of improving the simulation tool called NATO Reference Mobility Model (NRMM), used for simulating vehicles and assessing their mobility [3].

Part of this effort included inviting software developers to compare their simulation tools to empirical test data and the NRMM software. Chapter 8A "Tracked vehicle verification and validation" in [3] presents the results from a number of software developers simulating the Armored Personnel Carrier (APC) M113 on various terrain and obstacles. The participating teams and their software are listed in table 2.1 where each developer has been given a letter of identification, the same letter as in the NG-NRMM report.

Table 2.1: Teams and their software participating in the NG-NRMM report

Software Developer (Identification)	**Software**
Advanced Science and Automation Corporation (A)	IVRESS/DIS
FunctionBay (B)	RecurDyn
MSC Software (C)	ADAMS
University of Wisconsin – Madison (D)	Chrono
Vehicle Systems Development Corporation (E)	NTVPM/NWVPM

For the different benchmark events each team got assigned a maturity level based on their performance. A short description of each maturity level is described in table 2.2

Table 2.2: A short description of each maturity level in the NG-NRMM report

Maturity Level	Description
1	**DEMONSTRATION:** Demonstration of a correct implementation of a theoretically and conceptually consistent model.
2	**PARAMETER SENSITIVITY DEMONSTRATION:** Verification that performance change with a change in system parameter such as Gross Vehicle Weight (GVW) or terrain deformability is consistent with theory and physics principles.
3	**INDEPENDENT USER VERIFICATION:** Independent user demonstration and correlation to vendor results.
4	**CROSS CODE VERIFICATION:** Cross verification with another accepted mobility simulation code.
5	**CALIBRATION:** Calibration to a real vehicle test data set.
6	**VALIDATION:** Blind correlation to a real vehicle test data set.
7	**PARAMETER VARIATION VALIDATION:** Blind correlation to a real vehicle test data set with a change in system parameter(s).

2.4 Multibody Dynamics

A multibody system is a collection of interconnected rigid bodies. With the aim of describing the time evolution of such a system in a physically consistent way, we need to fomulate and solve the equations gouverning the dynamics of the system.

Let us start by introducing a single rigid body with a body fixed reference fame. The position of this frame can be located in an absolute reference frame by a vector $\boldsymbol{r} \in \mathbb{R}^3$ and the orientation of can be specified by a rotation matrix $\boldsymbol{A} \in \mathbb{R}^{3\times3}$. Thus the position $\boldsymbol{r}^P$ of an arbitrary point P on the body can be expressed as

$$\boldsymbol{r}^P = \boldsymbol{r} + \boldsymbol{A}\bar{\boldsymbol{s}}^P$$

where $\bar{\boldsymbol{s}}^P$ is the position of point p expressed in the local frame. From here on a bar $\bar{\cdot}$ over the vector will indicate that it is expressed in a local frame.

The velocity of the point P can be expressed as

$$\boldsymbol{v}^P = \dot{\boldsymbol{r}} + \dot{\boldsymbol{A}}\bar{\boldsymbol{s}}^P = \dot{\boldsymbol{r}} + \tilde{\boldsymbol{\omega}}\boldsymbol{s}^P$$

,

where $\tilde{\boldsymbol{\omega}}$ is the skew-symmetric matrix associated with the angular velocity $\boldsymbol{\omega}$ of the local frame. A skew-symmetric matrix $\tilde{\boldsymbol{a}}$ with an associated vector $\boldsymbol{a} = [a_x, a_y, a_z]$ is defined in [10] as

$$\tilde{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

. From here on a tilde $\tilde{}$ over a matrix will indicate that it is a skew-symmetrix matrix.

The acceleration of the point P can be expressed as

$$\boldsymbol{a}^P = \ddot{\boldsymbol{r}} + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\boldsymbol{s}^P + \dot{\tilde{\boldsymbol{\omega}}}\boldsymbol{s}^P$$

.
In a system with multiple rigid bodies one can restrict the relative movement between two bodies with constraints that act like joints. These constraints are expressed as algebraic functions of the position and orientatation of the bodies. There are a number of simple constraints, for example that the dot product between vector $\bar{\boldsymbol{a}}_i$ on body i and $\bar{\boldsymbol{a}}_j$ on body j equals the function $f(t)$. The algebraic constraint equation for such a constraint is expressed as

$$\Phi(i, \bar{\boldsymbol{a}}_i, j, \bar{\boldsymbol{a}}_j, f(t)) = \bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{A}_j \bar{\boldsymbol{a}}_j - f(t) = 0$$

and the equation equals zero when the constraint is fullfilled by the two bodies. A number of similar simple constraints can be combined to create more complex joints like a revolute or universal joint, for more information see appendix B.

To move towards the equations of motion of such a system we will rely on the concept of virtual work. Consider a virtual displacement $(\delta\boldsymbol{r}, \delta\bar{\boldsymbol{\pi}})$ of a rigid body, where $\delta\boldsymbol{r}$ is a virtual translation defined as an arbitrary infinitesimal translation of the body fixed local frame; $\delta\bar{\boldsymbol{\pi}}$ is a virtual rotation which is related to an arbitrary infinitesimal change of the orientation of the body as

$$\delta\tilde{\bar{\boldsymbol{\pi}}} = \boldsymbol{A}^T \delta\boldsymbol{A}.$$

The virtual work δW is the work produced by a force along a virtual displacement.

The virtual work of a single rigid body i under a virtual displacement $(\delta\boldsymbol{r}_i, \delta\bar{\boldsymbol{\pi}}_i)$ due to externally applied force and torque $\boldsymbol{F}_i^a$ and $\bar{\boldsymbol{n}}_i$ is

$$\delta W^a = \delta\boldsymbol{r}_i^T \boldsymbol{F}_i^a + \delta\bar{\boldsymbol{\pi}}_i^T \bar{\boldsymbol{n}}_i^a.$$

Let us assume that body i is constrained in some arbitrary way. These constraints/joints will cause reaction forces and torques $\boldsymbol{F}_i^r$ and $\bar{\boldsymbol{n}}_i^r$. The virtual work due to these are

$$\delta W^r = \delta\boldsymbol{r}_i^T \boldsymbol{F}_i^r + \delta\bar{\boldsymbol{\pi}}_i^T \bar{\boldsymbol{n}}_i^r.$$

We now invoke **D'Alembert's form of the principle of virtual work** that states that a system of rigid bodies is in a dynamic equilibrium if we include the work performed by

the inertial forces. The virtual work on body i produced by the inertial forces are

$$\delta W_i = \delta \boldsymbol{r}_i^T \ddot{\boldsymbol{r}}_i m_i + \delta \bar{\boldsymbol{\pi}}_i^T [\bar{\boldsymbol{J}}_i \dot{\bar{\boldsymbol{\omega}}}_i + \tilde{\bar{\boldsymbol{\omega}}}_i \bar{\boldsymbol{J}}_i \bar{\boldsymbol{\omega}}_i]$$

where $\boldsymbol{J}_i$ is the moment of intertia tensor for body i.

Let us continue by considering a system of nb rigid bodies constrained in an arbitrary way with nc algebraic constraints. As stated by D'Alembert's form of the principle of virtual work the total virtual work of such a system is zero and can be expressed as

$$\delta W = \sum_{i=1}^{nb} [\delta \boldsymbol{r}_i^T (m_i \ddot{\boldsymbol{r}}_{\boldsymbol{i}} - \boldsymbol{F}_i^a - \boldsymbol{F}_i^r) + \delta \bar{\boldsymbol{\pi}}_i^T (\bar{\boldsymbol{J}}_i \dot{\bar{\boldsymbol{\omega}}}_i + \tilde{\bar{\boldsymbol{\omega}}}_i \bar{\boldsymbol{J}}_i \bar{\boldsymbol{\omega}}_i - \bar{\boldsymbol{n}}_i^a - \bar{\boldsymbol{n}}_i^r])) = 0 \tag{2.1}$$

which is valid for any set of virtual displacement $(\delta \boldsymbol{r}_1, \delta \bar{\boldsymbol{\pi}}_1), \ldots, (\delta \boldsymbol{r}_{nb}, \delta \bar{\boldsymbol{\pi}}_{nb})$. Because (2.1) holds for all virtual displacements, the coefficients are neccesarily zero giving us the equations of motion for the system of nb rigid bodies

$$\begin{aligned} \boldsymbol{m} \ddot{\boldsymbol{r}} &= \boldsymbol{F}^a + \boldsymbol{F}^r \\ \bar{\boldsymbol{J}} \dot{\bar{\boldsymbol{\omega}}} &= \boldsymbol{\tau} + \bar{\boldsymbol{n}}^r \end{aligned} \tag{2.2}$$

where

$$\boldsymbol{m} = \begin{bmatrix} m_1 \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \ldots & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & m_2 \boldsymbol{I}_3 & \ldots & \boldsymbol{0}_{3\times3} \\ & \ddots & & \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \ldots & m_{nb} \boldsymbol{I}_3 \end{bmatrix}, \bar{\boldsymbol{J}} = \begin{bmatrix} \bar{\boldsymbol{J}}_1 & \boldsymbol{0}_{3\times3} & \ldots & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \bar{\boldsymbol{J}}_2 & \ldots & \boldsymbol{0}_{3\times3} \\ & \ddots & & \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \ldots & \bar{\boldsymbol{J}}_{nb} \end{bmatrix}$$

$$\boldsymbol{F}^a = \begin{bmatrix} \boldsymbol{F}_1^a \\ \vdots \\ \boldsymbol{F}_{nb}^a \end{bmatrix}, \boldsymbol{F}^r = \begin{bmatrix} \boldsymbol{F}_1^r \\ \vdots \\ \boldsymbol{F}_{nb}^r \end{bmatrix}, \bar{\boldsymbol{n}}^r = \begin{bmatrix} \bar{\boldsymbol{n}}_1^r \\ \vdots \\ \bar{\boldsymbol{n}}_{nb}^r \end{bmatrix}, \boldsymbol{\tau} = \begin{bmatrix} \bar{\boldsymbol{n}}_1^a - \tilde{\bar{\omega}}_1 \bar{\boldsymbol{J}}_1 \bar{\omega}_1 \\ \vdots \\ \bar{\boldsymbol{n}}_{nb}^a - \tilde{\bar{\omega}}_{nb} \bar{\boldsymbol{J}}_{nb} \bar{\omega}_{nb} \end{bmatrix}$$

This version of the equations of motion (2.2) is not perticularily useful for finding $\ddot{\boldsymbol{r}}$ and $\dot{\bar{\boldsymbol{\omega}}}$ since it contains the unkown reaction forces and torques. To get around this we rely on the **Principal of Virtual Work**, specifically the fact that the total virtual work of the constraint reaction forces will be zero as long as the virtual displacements are consistent with the constraints [10]. Or in mathematical notation:

$$\delta \boldsymbol{r}^T \boldsymbol{F}^r + \delta \bar{\boldsymbol{\pi}}^T \bar{\boldsymbol{n}}^r = 0$$

if

$$\boldsymbol{\Phi}_{\boldsymbol{r}} \delta \boldsymbol{r} + \bar{\boldsymbol{\Pi}}(\boldsymbol{\Phi}) \delta \bar{\boldsymbol{\pi}} = \boldsymbol{0} \tag{2.3}$$

where $\boldsymbol{\Phi}_{\boldsymbol{r}}$ is the partial derivative of the constraints with respect to position and $\bar{\boldsymbol{\Pi}}(\boldsymbol{\Phi})$ is the coefficient matrix that is multiplied with $\bar{\boldsymbol{\omega}}$ in the expression of the time derivative $\dot{\boldsymbol{\Phi}}$, see appendix x for more details. Furthermore, since the total virtual work (2.1) is still zero we have

$$\delta W = \delta \boldsymbol{r}^T (\boldsymbol{m} \ddot{\boldsymbol{r}} - \boldsymbol{F}^a) + \delta \bar{\boldsymbol{\pi}}^T (\bar{\boldsymbol{J}} \dot{\bar{\boldsymbol{\omega}}} - \boldsymbol{\tau}) = 0 \tag{2.4}$$

given that (2.3) holds, where

$$\delta \boldsymbol{r} = \begin{bmatrix} \delta \boldsymbol{r}_1 \\ \vdots \\ \delta \boldsymbol{r}_{nb} \end{bmatrix}, \delta \bar{\boldsymbol{\pi}} = \begin{bmatrix} \delta \bar{\boldsymbol{\pi}}_1 \\ \vdots \\ \delta \bar{\boldsymbol{\pi}}_{nb} \end{bmatrix}, \bar{\boldsymbol{n}}^a = \begin{bmatrix} \bar{\boldsymbol{n}}_1^a \\ \vdots \\ \bar{\boldsymbol{n}}_{nb}^a \end{bmatrix}$$

.

We are now ready to use the **Lagrange Multiplier theorem** which is defined in [10] as: Given a vector $\boldsymbol{b}$ of n constants, a vector $\boldsymbol{x}$ of n variables and a constant $m \times n$ matrix $\boldsymbol{A}$. If

$$\boldsymbol{b}^T \boldsymbol{x} = 0$$

hold for all $\boldsymbol{x}$ that satisfy

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$$

then there exists a vector $\boldsymbol{\lambda}$ of m Lagrange multipliers such that

$$\boldsymbol{b}^T \boldsymbol{x} + \boldsymbol{\lambda}^T \boldsymbol{A}\boldsymbol{x} = 0$$

Rephrasing (2.3) and (2.4) as

$$\delta W = \begin{bmatrix} \delta \boldsymbol{r} \\ \delta \bar{\boldsymbol{\pi}} \end{bmatrix}^T \cdot \begin{bmatrix} \boldsymbol{m}\ddot{\boldsymbol{r}} - \boldsymbol{F}^a \\ \bar{\boldsymbol{J}}\dot{\bar{\boldsymbol{\omega}}} - \boldsymbol{\tau} \end{bmatrix} = 0$$

holds for all $[\delta \boldsymbol{r}, \delta \bar{\boldsymbol{\pi}}]^T$ that satisfy

$$\begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{r}} \\ \bar{\boldsymbol{\Pi}}(\boldsymbol{\Phi}) \end{bmatrix}^T \cdot \begin{bmatrix} \delta \boldsymbol{r} \\ \delta \bar{\boldsymbol{\pi}} \end{bmatrix} = \boldsymbol{0}.$$

Then according to the Lagrange Multiplier theorem, there is a vector of nc scalars

$$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{nc}]^T$$

such that

$$\begin{bmatrix} \boldsymbol{m}\ddot{\boldsymbol{r}} - \boldsymbol{F}^a \\ \bar{\boldsymbol{J}}\dot{\bar{\omega}} - \tau \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{r}}{}^T \\ \bar{\boldsymbol{\Pi}}^T(\boldsymbol{\Phi}) \end{bmatrix} \boldsymbol{\lambda} = \boldsymbol{0} \tag{2.5}$$

which together with the constraint equations

$$\boldsymbol{\Phi} = \boldsymbol{0} \tag{2.6}$$

form the constrained equations of motion for the system at a specific time. These equations of motion comprise a system of Differential Algebraic Equations (DAE), due to the mix of differential and algebraic equations.

2.4.1 Equations of Motion in Chrono

While (2.5) and (2.6) form the essential equations used in Chrono, the implementation of these equations in the code and in the documentation differs. This section will formulate the equations as they are presented in [11] and [12].

We start by introducing the following generalized coordinates and velocities for nb rigid bodies as used in Chrono

$$\boldsymbol{q} = [\boldsymbol{r}_1^T, \boldsymbol{\rho}_1^T, ..., \boldsymbol{r}_{nb}^T, \boldsymbol{\rho}_{nb}^T]^T \in \mathbb{R}^{7nb},$$

$$\boldsymbol{v} = [\dot{\boldsymbol{r}}_1^T, \bar{\boldsymbol{\omega}}_1^T, ..., \dot{\boldsymbol{r}}_{nb}^T, \bar{\boldsymbol{\omega}}_{nb}^T]^T \in \mathbb{R}^{6nb}.$$

Here $\boldsymbol{q}$ is the spatial configuration where $\boldsymbol{r}_i \in \mathbb{R}^3$ is the position of the center of mass for the i-th body, expressed in the global frame. The orientation of a body – also expressed in the global frame – is parameterized by a quaternion on the hypersphere of unit-length quaternions, see appendix x for how quaternions are related to rotation and orientation. Thus the orientation $\boldsymbol{\rho}_i \in \mathbb{R}^4$ of the i-th body is described by 4 scalars. The velocity component $\boldsymbol{v}$ of the system state includes the translational velocity $\dot{\boldsymbol{r}}_i \in \mathbb{R}^3$ for every i-th body, expressed in the absolute reference. The angular velocity of the i-th body $\bar{\boldsymbol{\omega}}_i \in \mathbb{R}^3$ on the other hand is expressed in its body fixed reference frame.

The time derivative of the generalized positions can be obtained from the generalized velocities as $\dot{\boldsymbol{q}} = \mathbf{L}(\boldsymbol{q})\boldsymbol{v}$ where $\mathbf{L}$ is a linear map that transforms the angular velocites into a quaternion derivative using the formula $\dot{\boldsymbol{\rho}}_i = \frac{1}{2}\boldsymbol{G}^T(\boldsymbol{\rho}_i)\bar{\boldsymbol{\omega}}_i$ [12]. Where $\boldsymbol{G}(\boldsymbol{\rho})$ is defined in [10] as

$$\boldsymbol{G}(\boldsymbol{\rho}) = \begin{bmatrix} -\rho_1 & \rho_0 & \rho_3 & -\rho_2 \\ -\rho_2 & -\rho_3 & \rho_0 & \rho_1 \\ -\rho_3 & \rho_2 & -\rho_1 & \rho_0 \end{bmatrix}$$

leading to $\mathbf{L}$ having the form

$$\mathbf{L}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \frac{1}{2}\boldsymbol{G}^T(\boldsymbol{\rho}_1) & \dots & \mathbf{0}_{3\times3} \\ & \ddots & \ddots & \\ \mathbf{0}_{4\times3} & \dots & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \dots & \mathbf{0}_{4\times3} & \frac{1}{2}\boldsymbol{G}^T(\boldsymbol{\rho}_{nb}) \end{bmatrix}_{7nb\times6nb}.$$

The system of rigid bodies is constrained by nc basic algebraic constraints denoted

$$\boldsymbol{C}(\boldsymbol{q}, t) = [C_1(\boldsymbol{q}, t), \dots, C_{nc}(\boldsymbol{q}, t)]$$

The constraints are differentiated to obtain the Jacobian $\nabla_q C_i = [\partial C_i / \partial \boldsymbol{q}]$. As is done in [11] and [12] we will abbreviate

$$\nabla_q C_i = \nabla_q C_i \mathbf{L}(\boldsymbol{q}) \in \mathbb{R}^{6nb}$$

to end up with the correct dimensions in the final equations of motion.

Introducing the generalized force $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t)$ and mass-inerta matrix $\boldsymbol{M}$

$$\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) = \begin{bmatrix} \boldsymbol{F}_1^a \\ \bar{\boldsymbol{n}}_1^a \\ \vdots \\ \boldsymbol{F}_{nb}^a \\ \bar{\boldsymbol{n}}_{nb}^a \end{bmatrix}_{6nb}, \boldsymbol{M} = \begin{bmatrix} m_1 \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \dots & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \bar{\boldsymbol{J}}_1 & \dots & \boldsymbol{0}_{3\times3} \\ & \ddots & \ddots & \\ \boldsymbol{0}_{3\times3} & \dots & m_{nb}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \dots & \boldsymbol{0}_{3\times3} & \bar{\boldsymbol{J}}_{nb} \end{bmatrix}_{6nb\times6nb}$$

We can now formulate the equations of motion as

$$\dot{\boldsymbol{q}} = \mathbf{L}(\boldsymbol{q})\boldsymbol{v}, \tag{2.7}$$
$$\boldsymbol{C}(\boldsymbol{q}, t) = \boldsymbol{0},$$
$$\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) - \sum_{i=1}^{nc} \nabla_q C_i^T \lambda_i \tag{2.8}$$

2.4.2 Time Stepping

With the equations of motion formulated we can move on to solving them at discrete time steps and advancing the system forwards in time. All of the time stepping methods explained below perform some sort of operation of the sort $\boldsymbol{q}_{new} = \boldsymbol{q} + h\boldsymbol{v}$ where h is the size of a time step. However since the size of $\boldsymbol{q}$ and $\boldsymbol{v}$ differ this operation will from here on be evaluated as

$$\boldsymbol{q}_{new} = \exp(h\boldsymbol{v})\boldsymbol{q} = \begin{bmatrix} \boldsymbol{r}_1 + h\dot{\boldsymbol{r}}_1 \\ \boldsymbol{\rho}_1 \exp([0, \frac{1}{2}\bar{\boldsymbol{\omega}}_1 h]) \\ \vdots \\ \boldsymbol{r}_{nb} + h\dot{\boldsymbol{r}}_{nb} \\ \boldsymbol{\rho}_{nb} \exp([0, \frac{1}{2}\bar{\boldsymbol{\omega}}_{nb} h]) \end{bmatrix} \tag{2.9}$$

as explained in [11], where $\exp(\boldsymbol{\rho})$ is the quaternion exponential map defined in appendix A.

Euler Implicit

We will begin to approch the time stepping methods by looking at the Euler implicit method of time integration *without* constraints, defined as

$$\boldsymbol{q}^{l+1} = \boldsymbol{q}^l + h\boldsymbol{v}^{l+1}$$
$$\boldsymbol{v}^{l+1} = \boldsymbol{v}^l + h\boldsymbol{a}^{l+1}$$

where h is the size of the time step and $\boldsymbol{a}$ is related to the external forces $\boldsymbol{f}$ as $\boldsymbol{f}(\boldsymbol{q}^{l+1}, \boldsymbol{v}^{l+1}, t^{l+1}) = \boldsymbol{M}\boldsymbol{a}^{l+1}$. Let us establish the abbreviation $\boldsymbol{f}(\boldsymbol{q}^{l+1}, \boldsymbol{v}^{l+1}, t)$ as $\boldsymbol{f}^{l+1}$.

For our purposes we instead define the residuals for the above equations as

$$\boldsymbol{G}_q = \boldsymbol{q}^{l+1} - \boldsymbol{q}^l - h\boldsymbol{v}^{l+1}$$
$$\boldsymbol{G}_v = \boldsymbol{M}(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l) - h\boldsymbol{f}^{l+1}.$$

The goal is to find the $\boldsymbol{q}^{l+1}$ and $\boldsymbol{v}^{l+1}$ that satisfy $\boldsymbol{G}_q = 0, \boldsymbol{G}_v = 0$, which is done using the Newton-Raphson method.

A quick refresher on the Newton-Raphson (NR) method. The NR method is an iterative algorithm for finding the roots of a real-valued function. For a one-dimensional function $f(x)$ we would have an iterative step as follows:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Extending it to a vector valued function $\boldsymbol{F}(\boldsymbol{x})$ we instead get

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - J_F(\boldsymbol{x}_n)^{-1}\boldsymbol{F}(\boldsymbol{x}_n), \tag{2.10}$$

where J_F^{-1} is the inverse of the Jacobian of F. Calculating the inverse Jacobian is often an ill-conditioned problem, to avoid this it is common to multiply (2.10) with the Jacobian from the left and instead get a system of equations. Replacing $\boldsymbol{F}$ with $\boldsymbol{G} = [\boldsymbol{G}_q, \boldsymbol{G}_v]^{\mathrm{T}}$ and $\boldsymbol{x}$ with $[\boldsymbol{q}, \boldsymbol{v}]$ and framing the problem as a system of equations we get

$$J_G\left(\begin{bmatrix}\boldsymbol{q}_n^{l+1}\\ \boldsymbol{v}_n^{l+1}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{q}_{n+1}^{l+1} - \boldsymbol{q}_n^{l+1}\\ \boldsymbol{v}_{n+1}^{l+1} - \boldsymbol{v}_n^{l+1}\end{bmatrix} = -\begin{bmatrix}\boldsymbol{G_q}\\ \boldsymbol{G_v}\end{bmatrix}.$$

We introduce the following notation $J_n^l = J_G([\boldsymbol{q}_n^l, \boldsymbol{v}_n^l]^{\mathrm{T}})$ and $\Delta\boldsymbol{q}^l = \boldsymbol{q}_{n+1}^l - \boldsymbol{q}_n^l$, where

$$J_n^l = \begin{bmatrix}\frac{\partial \boldsymbol{G_q}}{\partial \boldsymbol{q}} & \frac{\partial \boldsymbol{G_q}}{\partial \boldsymbol{v}}\\ \frac{\partial \boldsymbol{G_v}}{\partial \boldsymbol{q}} & \frac{\partial \boldsymbol{G_v}}{\partial \boldsymbol{v}}\end{bmatrix}_n^l = \begin{bmatrix} I & -hI\\ -h\nabla_q\boldsymbol{f}^{l+1} & \boldsymbol{M} - h\nabla_v\boldsymbol{f}^{l+1}\end{bmatrix}.$$

Using the approximation

$$\Delta\boldsymbol{q}^{l+1} = h\Delta\boldsymbol{v}^{l+1} \tag{2.11}$$

and evaluating the matrix multiplications we end up with the NR step procedure:

$$\begin{aligned}(\boldsymbol{M} - h^2\nabla_q\boldsymbol{f}^{l+1} - h\nabla_v\boldsymbol{f}^{l+1})\Delta\boldsymbol{v}^{l+1} &= (\boldsymbol{v}^l - \boldsymbol{v}^{l+1})\boldsymbol{M} + h\boldsymbol{f}^{l+1}\\ \boldsymbol{v}_{n+1}^{l+1} &= \boldsymbol{v}_n^{l+1} + \Delta\boldsymbol{v}^{l+1}\\ \boldsymbol{q}^{l+1} &= \boldsymbol{q}^l + h\boldsymbol{v}_{n+1}^{l+1},\end{aligned}$$

which can be repeated at each time step until the desired precision is reached.

Adding constraints
When the constraints $\boldsymbol{C}(\boldsymbol{q}, t) = 0$ are added we redefine the residual functions as

$$
\begin{aligned}
\boldsymbol{G}_q &= \boldsymbol{q}^{l+1} - \boldsymbol{q}^l - h\boldsymbol{v}^{l+1} \\
\boldsymbol{G}_v &= \boldsymbol{M}(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l) - h\boldsymbol{f}^{l+1} - h\nabla_q \boldsymbol{C}\boldsymbol{\lambda}^{l+1} \\
\boldsymbol{G}_c &= \boldsymbol{C}(\boldsymbol{q}^{l+1}, t^{l+1})
\end{aligned}
$$

giving us the following NR iteration:

$$
\begin{bmatrix} I & -hI & 0 \\ -h\nabla_q \boldsymbol{f}^{l+1} & \boldsymbol{M} - h\nabla_v \boldsymbol{f}^{l+1} & -h\nabla_q \boldsymbol{C} \\ \nabla_q \boldsymbol{C}^{\mathrm{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q}^{l+1} \\ \Delta\boldsymbol{v}^{l+1} \\ \Delta\boldsymbol{\lambda}^{l+1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{G_q} \\ \boldsymbol{G_v} \\ \boldsymbol{G_c} \end{bmatrix}.
$$

Which when applying (2.11) and evaluating gives us

$$
\begin{bmatrix} (\boldsymbol{M} - h^2\nabla_q \boldsymbol{f}^{l+1} - h\nabla_v \boldsymbol{f}^{l+1}) & \nabla_q \boldsymbol{C} \\ \nabla_q \boldsymbol{C}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{v}^{l+1} \\ -h\Delta\boldsymbol{\lambda}^{l+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}(\boldsymbol{v}^l - \boldsymbol{v}^{l+1}) + h\boldsymbol{f}^{l+1} + h\nabla_q \boldsymbol{C}\boldsymbol{\lambda}^{l+1} \\ -\frac{\boldsymbol{C}(\boldsymbol{q}^{l+1}, t^{l+1})}{h} \end{bmatrix} \tag{2.12}
$$

$$
\begin{aligned}
\boldsymbol{v}_{n+1}^{l+1} &= \boldsymbol{v}_n^{l+1} + \Delta\boldsymbol{v}^{l+1} \\
\boldsymbol{\lambda}_{n+1}^{l+1} &= \boldsymbol{\lambda}_n^{l+1} + \Delta\boldsymbol{\lambda}^{l+1} \\
\boldsymbol{q}^{l+1} &= \boldsymbol{q}^l + h\boldsymbol{v}_{n+1}^{l+1}
\end{aligned}
$$

where $\boldsymbol{C}^{l+1} = \boldsymbol{C}(\boldsymbol{q}^{l+1}, t^{l+1})$. This is the time stepping method implemented in Chrono under the name `ChTimestepperEulerImplicit` [13].

Euler Implicit Linearized

To arrive at the default time stepping scheme in Chrono, called `ChTimestepperEulerImplicitLinearized` [14], the following approximations are done:

$$
\begin{aligned}
\Delta\boldsymbol{v}^{l+1} &\approx \boldsymbol{v}^{l+1} \\
\boldsymbol{f}^{l+1} &\approx \boldsymbol{f}^l \\
\Delta\boldsymbol{\lambda}^{l+1} &\approx \boldsymbol{\lambda}^{l+1} \\
\boldsymbol{C}^{l+1} &\approx \boldsymbol{C}^l + \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Delta\boldsymbol{q} + \frac{\partial \boldsymbol{C}}{\partial t}\Delta t.
\end{aligned}
$$

Using the approximations above and only performing a single NR step we get the time stepping procedure as

$$
\begin{bmatrix} (\boldsymbol{M} - h^2\nabla_q \boldsymbol{f}^{l+1} - h\nabla_v \boldsymbol{f}^{l+1}) & \nabla_q \boldsymbol{C} \\ \nabla_q \boldsymbol{C}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{l+1} \\ -h\boldsymbol{\lambda}^{l+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}\boldsymbol{v}^l + h\boldsymbol{f}^l \\ -\frac{\boldsymbol{C}^l}{h} - \frac{\partial \boldsymbol{C}}{\partial t} \end{bmatrix} \tag{2.13}
$$

$$
\boldsymbol{q}^{l+1} = \boldsymbol{q}^l + h\boldsymbol{v}^{l+1}
$$

Euler Implicit Projected
An alternative time stepping scheme in Chrono, called `ChTimestepperEulerImplicitProjected` [15]. Similarly to the Euler Implicit Linearized time stepper it begins by performing a single NR step with the same approximations while omitting the $\frac{C^l}{h}$ constraint term.

$$\begin{bmatrix} (\boldsymbol{M} - h^2 \nabla_q \boldsymbol{f}^{l+1} - h \nabla_v \boldsymbol{f}^{l+1}) & \nabla_q \boldsymbol{C} \\ \nabla_q \boldsymbol{C}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{l+1} \\ -h\boldsymbol{\lambda}^{l+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}\boldsymbol{v}^l + h\boldsymbol{f}^l \\ -\frac{\partial \boldsymbol{C}}{\partial t} \end{bmatrix} \tag{2.14}$$
$$\boldsymbol{q}^{l+1} = \boldsymbol{q}^l + h\boldsymbol{v}^{l+1}$$

To rectify the potential constraint violations this causes the positions are forced into a non-violating state by performing a NR step with only the constraints and positions as

$$\begin{bmatrix} \boldsymbol{M} & \nabla_q \boldsymbol{C} \\ \nabla_q \boldsymbol{C}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{q} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ -\boldsymbol{C} \end{bmatrix} \tag{2.15}$$
$$\boldsymbol{q}^{l+1} = \boldsymbol{q}^l + \Delta \boldsymbol{q}$$

Sovlers
A common feature between the time stepping methods explained above, and in fact all methods implemented in Chrono, is that they require solving one or more linear systems at every time step. Looking at (2.12), (2.13), (2.14) and (2.15) one sees that the systems share the common structure of

$$\begin{bmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

which is known as a saddle point problem, a common problem in computational science [16]. Chrono offers multiple alternatives to solving this system.

2.4.3 Contact Forces

So far we have discussed the relation between applied forces and constraint forces and how to ensure the constraints are enforced during a simulation. However we have not touched the subject of contact forces that occur when rigid bodies collide. Chrono offers two distinctly different ways of dealing with contacts which they call smooth contact (SMC) and non-smooth contact (NSC). In this thesis the SMC method is almost exclusively used and is thus the focus.

SMC uses the DAE based approach explained above and it allows for some interpenetration between bodies, which can be thought of as treating all bodies as locally deformable. This interpenetration is then used to calculate the reaction forces $\boldsymbol{F}_n$ and $\boldsymbol{F}_t$ in the normal and tangential directions, based on a chosen model. Using Hertzian contact theory we get the following contact forces

$$\begin{aligned} \boldsymbol{F}_n &= \sqrt{\bar{R}\delta_n}(k_n \boldsymbol{\delta}_n - \gamma_n \bar{m} \boldsymbol{v}_n) \\ \boldsymbol{F}_t &= \sqrt{\bar{R}\delta_n}(-k_t \boldsymbol{\delta}_t - \gamma_t \bar{m} \boldsymbol{v}_t) \end{aligned} \tag{2.16}$$

where $\boldsymbol{\delta} = \boldsymbol{\delta_n} + \boldsymbol{\delta_t}$ is the local penetration of the involved bodies i and j who have an effective mass $\bar{m} = m_i m_j/(m_i + m_j)$, effective radius of curvature $\bar{R} = R_i R_j/(R_i + R_j)$ and a relative velocity at the contact point $\boldsymbol{v} = \boldsymbol{v}_n + \boldsymbol{v}_t$. The normal and tangential stiffness and damping coefficients k_n, k_t, γ_n and γ_t can either be specified by the user or calculated based on material properties.

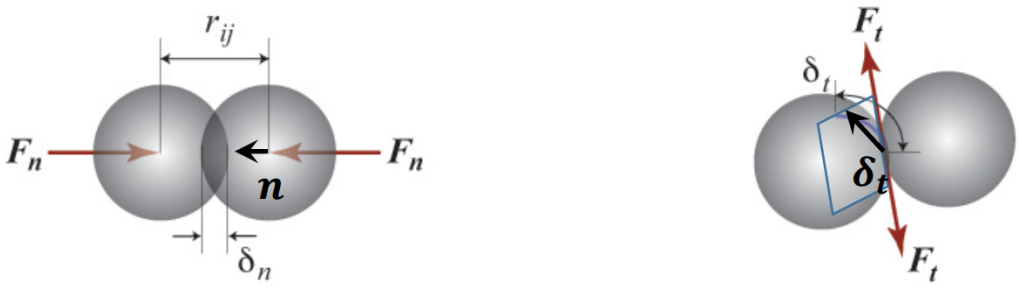

Figure 2.1: A visual representation of the contact displacement in the normal direction (left) and tangential direction (right). [2]

The tangential contact displacement $\boldsymbol{\delta}_t$ will always lie in the contact plane, perpendicular to the normal displacement and velocity, δ_n and $\boldsymbol{v}_n$. However it must not necessarily be parallel to the tangential velocity $\boldsymbol{v}_t$, for example when a tangential contact force must act on a body to support *static* friction. This is achieved by updating $\boldsymbol{\delta}_t$ at every time step two bodies are in contact and projecting $\boldsymbol{\delta}_t$ on the contact plane as

$$\begin{aligned} \boldsymbol{\delta}_t^* &= \sum \boldsymbol{v}_t \Delta t \\ \boldsymbol{\delta}_t &= \boldsymbol{\delta}_t^* - (\boldsymbol{\delta}_t^* \cdot \boldsymbol{n})\boldsymbol{n}, \end{aligned}$$

see figure 2.4.3 for a visual representation. Finally, before the tangential force is applied, the Coulomb friction law is applied ensuring that $|\boldsymbol{F}_t| \leq \mu |\boldsymbol{F}_n|$.

The normal and tangential stiffness coefficients in are derived as

$$k_n = \frac{4}{3} \bar{E} \sqrt{\bar{R}\delta_n} \tag{2.17}$$

$$k_t = 8\bar{G}\sqrt{\bar{R}\delta_n}, \tag{2.18}$$

when using material properties rather than explicitly defining them. Here $\bar{E}$ is the effective Young modulus and $\bar{G}$ the effective shear modulus. These properties are calculated as

$$\begin{aligned} \bar{E} &= \left(\frac{1-\nu_i^2}{E_i} + \frac{1-\nu_j^2}{E_j} \right)^{-1} \\ \bar{G} &= \left(\frac{2(2+\nu_i)(1-\nu_i)}{E_i} + \frac{2(2+\nu_j)(1-\nu_j)}{E_j} \right)^{-1}, \end{aligned} \tag{2.19}$$

where E_i and E_j is the Young modulus of the materials of the two bodies i and j in contact and ν_i and ν_j their Poisson's ratios. More details on contact forces, materials and coefficients in Chrono can be found in [17].

When using the SMC method the calculated contact forces are treated as any other external forces and is added to $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t)$ in (2.8). Meanwhile, the NSC method incorporates the contacts as a second type of constraint, leading to different types of governing equations than those of (2.7) to (2.8). The NSC method allows for completely rigid bodies, without the local deformations used in the SMC method. A detailed explanation of the NSC method can be found in [18].

3
Modeling

3.1 Creating a Parameterized Vehicle model in `Chrono`

Section 3.1.1 lays out the structure of vehicle simulations using the `Chrono::Vehicle` module, and introduces some of the vocabulary needed to understand the next section. Section 3.1.2 explains how a dynamic vehicle model was created using the `Chrono::Vehicle` module.

To clarify the syntax of the thesis, for every keyword, class name or any word present in the code `this font will be used`.

3.1.1 `Chrono::Vehicle`

`Chrono::Vehicle` is the module that provides the code for the modeling, simulation, and visualization of vehicles. The code consists of a handful of systems, which are separated functional blocks of code that communicate with each other in predefined ways. The relevant systems for this thesis are: `Driver`, `Powertrain`, `Terrain` and `Vehicle`. A diagram showing the communication between the systems is shown in figure 3.1, for example the `Driver` communicates with the `Powertrain` *only* by sending information about the throttle input. A more detailed overview of the vehicle module can be found at [19].

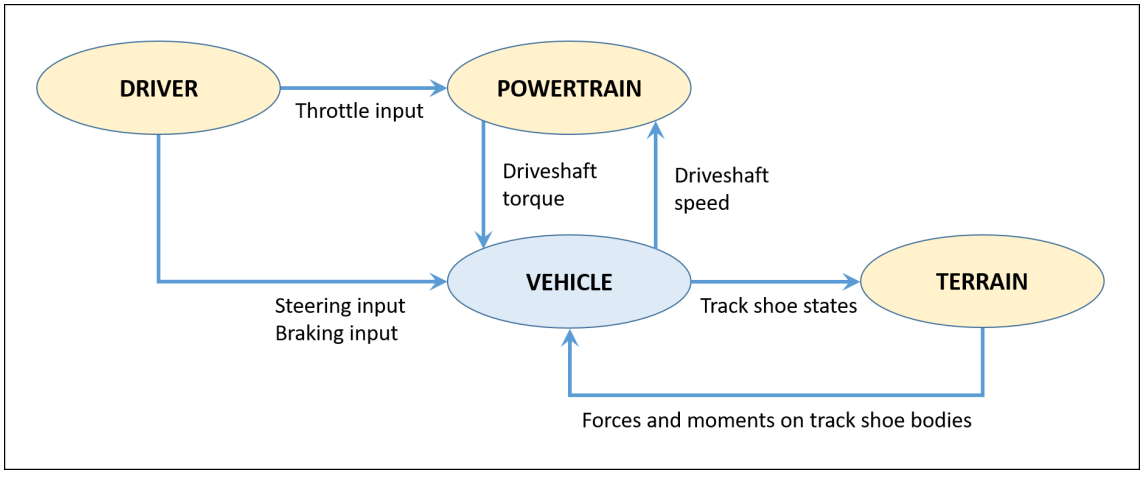

Figure 3.1: An overview of the communication between the independent systems for a tracked vehicle, image from [19]

The vehicle system is in itself made up of subsystems with distinct functional roles, for example the subsystem `ChTrackAssembly` whose role is, among other things, to initialize and assemble the track-shoes. These subsystems are abstract base classes that define the basic modeling elements and overarching functionality using both virtual and non-virtual functions as well as class attributes. A subsystem is then further specified through what Chrono calls a *template*, not to be confused with the C++ keyword `template`, for example `ChTrackAssemblySinglePin` is a *template*. A *template* is an abstract child class derived from a subsystem base class, *templates* define some virtual functions and variables from the type class but not all, there are typically multiple *templates* per subsystem. The final concrete class is a user-defined class that inherits from a *template* and satisfies all virtual functions. Figure 3.3 shows a visual example of the structure.

A diagram showing some of the subsystems that make up a tracked vehicle can be seen in figure 3.2. It should be noted that the track-assembly subsystem is composed of multiple subsystems, among them the sprocket, idler and suspension. The `Vehicle` system communicates indirectly with these subsystems through the track-assembly subsystem.

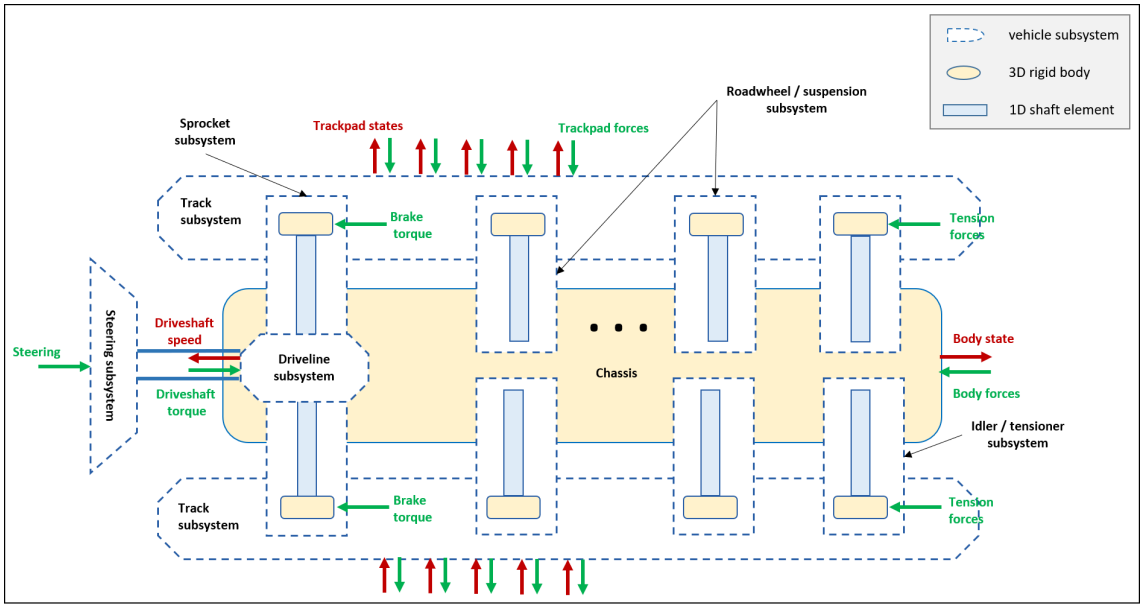

Figure 3.2: An overview of some the subsystems that make up a tracked vehicle, image from [20]

3.1.2 Making the Model Dynamic

Dynamically (during runtime) varying the parameters that define a vehicle in Chrono is not as straightforward as one might initially believe. The `Chrono::Vehicle` module does offer fully parameterized models where close to every aspect, down to the Young modulus of the guide pin on the track shoes, can be specified and different *templates* of subsystems can be swapped with relative ease. However, many of these parameters are not meant to be changed once they have been specified, due to the often complex interaction

and mutual dependence between parameters. Simply put, most parameters – if changed – will break the model, and for the parameters that can be altered without breaking the model, there are no callable functions to change them. This static structure and the limitations that follow are expressed in the Chrono forum as well [21], suggesting that it is not an uncommon problem to encounter.

Thus, a modifiable vehicle model would firstly allow access to its parameters and secondly keep track of the relationships between its defining parameters to ensure the desired change is valid. To allow more extensive modifications the model would have to change not only the specified parameter, but also the related parameters in a way that keeps the model intact.

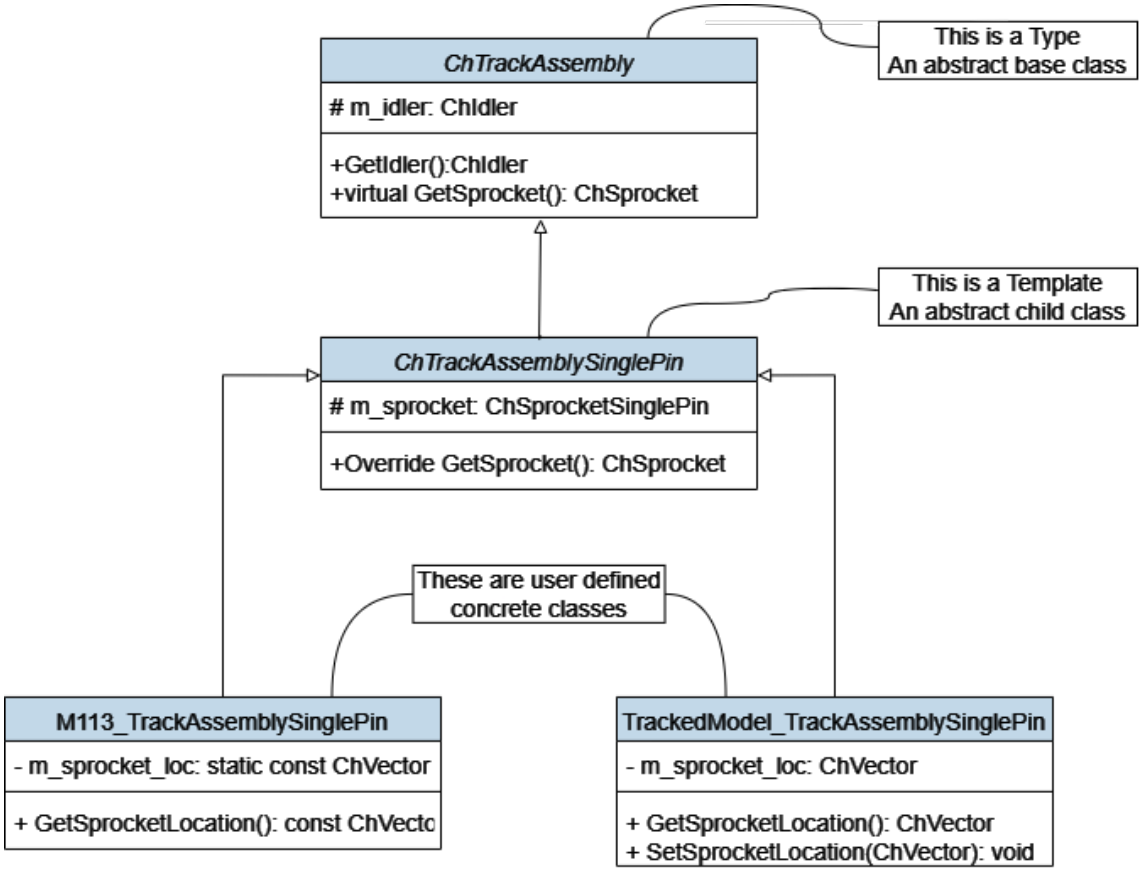

Figure 3.3: A UML Class diagram displaying the class hierarchy used in `Chrono::Vehicle`, this is a temporary image

Figure 3.3 illustrates the class hierarchy used in `Chrono::Vehicle`, where the track-assembly was chosen as an example. Looking at the *template* class `ChTrackAssemblySinglePin` we see that there is no functionality for changing the sprocket location. In the provided demo models, for example the M113 vehicle, the sprocket location is defined as a `static const vector`, as a consequence this attribute has to be "hardcoded". This is solved by creating a new concrete class implementing the single-pin *template*. In this class, called `TrackedModel_TrackAssemblySinglePin`, the `static` and `const` properties

of the sprocket location attribute can be removed and the desired functionality can be added.

Unfortunately this is not enough to be able to change the sprocket position. Another problem arises as the subsystems are integrated into the `Vehicle` system. Similarly to how `ChTrackAssembly` is the base class for the track-assembly subsystem, `ChTrackedVehicle` is the base class for the vehicle system. As seen in figure 3.3, `ChTrackAssembly` has a virtual function for returning a shared pointer to the sprocket, with return type of base class `ChSprocket`. In a similar way, `ChTrackedVehicle` has a track-assembly attribute of type `ChTrackAssembly` and a virtual function for returning it. This means that any concrete tracked vehicle class, such as `TrackedModel_vehicle` will only be able to return the track-assembly base class, due to the need for overriding virtual functions to have the same signature as defined in the parent class.

This problem can be solved in multiple ways. One could potentially alter the Chrono source code for the base classes to include the desired functionality. This however causes a chain of errors needing to be addressed and complicates any potential version updates of Chrono. Instead, another option was chosen that defines a manager class called `TrackedModel` keeping track of all concrete classes, such as `TrackedModel_TrackAssemblySinglePin` and `TrackedModel_vehicle`. Parameters can then be modified through the manager class. When it is time to initialize the vehicle, the manager class creates copies of the concrete classes and converts them into their base classes (where the functionality is lost but the specified parameter values remain) and defines a vehicle. When a parameter needs to be changed, the manager class simply deletes the current vehicle and defines a new one with the desired parameters.

3.2 Modelling the M113 APC

Figure 3.4: An M113 in the wild. Image by Expert Infantry, CC BY 2.0, via Wikimedia Commons

The vehicle model's base configuration will be the M113 APC used in the NG-NRMM report [3]. In addition to the data found in the NG-NRMM report, the technical report "NG-NRMM Phase I Benchmarking: Chrono Tracked Vehicle Simulation Results Summary " [22] produced by the Chrono development team will be referenced. As in [22] the M113 model provided by the `Chrono::Vehicle` module will be used, modified as needed to align with the data provided in the NG-NRMM report. It should be stated that the results presented by the Chrono development team in [22] are not the same results presented by the Chrono development team in the NG-NRMM report [3].

The following chapters will specify all changes done to the default M113 using the single pin track assembly. Any property of the vehicle not specified here will be the default value for the M113 single pin track in Chrono version 8.0.0.

3.2.1 Mass, inertia and size

The following values were taken from [3] when applicable, in second hand calculated as described in [22]. The remaining parameters were left as default. The origin of the chassis coordinate system is the sprocket center.

Table 3.1: Chassis properties

Mass	7804 [kg]
Inertia xx	13983.08 [kgm^2]
Inertia yy	27287.61 [kgm^2]
Inertia zz	28805.57 [kgm^2]
Center of gravity x	-1.98 [m]
Center of gravity y	0.48 [m]
Driver position x	-1.73 [m]
Driver position y	0.5 [m]
Driver position z	0.508 [m]

Table 3.2: Sprocket properties

Mass	66.21 [kg]
Inertia xx	1.07 [kgm^2]
Inertia yy	1.52 [kgm^2]
Inertia zz	1.07 [kgm^2]

Table 3.3: Idler wheel properties

Mass	38.2 [kg]
Radius	0.219 [m]
Inertia xx	0.52 [kgm^2]
Inertia yy	0.92 [kgm^2]
Inertia zz	0.52 [kgm^2]

Table 3.4: Road wheel properties

Mass	74.10 [kg]
Radius	0.305 [m]
Inertia xx	1.83 [kgm^2]
Inertia yy	3.45 [kgm^2]
Inertia zz	1.83 [kgm^2]

Table 3.5: Suspension arm properties

Mass	29.32 [kg]
Inertia xx	0.0132 [kgm^2]
Inertia yy	0.257 [kgm^2]
Inertia zz	0.257 [kgm^2]

Table 3.6: Track shoe properties

Mass	8.55 [kg]
Pitch	0.154 [m]
Inertia xx	0.1562 [kgm^2]
Inertia yy	0.0318 [kgm^2]
Inertia zz	0.1562 [kgm^2]

3.2.2 Suspension

The default suspension arm was altered to be straight, as this is what is shown in [22]. Every road wheel is attached to a torsion bar, the rebound and jounce limits for the torsion bars are enforced by significantly increasing the spring stiffness at the limits. The left side torsion arm pivot points are shown in table 3.7, the suspension arm pivot points on the right side are shifted 0.1016 m backwards in accordance to [3]. It should be noted that when using the `Chrono::Vehicle` module the wheel centers are specified rather than the suspension arm pivot points, meaning one has to specify the torsion arm pivot points by the hardpoints defined in the `ChTranslationalDamperSuspension` class. The spring properties are shown in table 3.8.

Table 3.7: Torsion arm pivot points on the left side of the chassis, origin is placed at the sprocket center. Positive x-direction is forward and positive z-direction is up.

Wheel nr	**x-coordinate [m]**	**z-coordinate [m]**
1	-0.3969	-0.0873
2	-1.0636	-0.0873
3	-1.7304	-0.0873
4	-2.3971	-0.0873
5	-3.0639	-0.0873

Vertical linear dampers are fitted on the wheel center of the first and last wheel pair as described in [3]. The dampers are modeled according to the assumptions presented in the NG-NRMM report [3], translated to SI units in table 3.9. The report does not specify jounce mode and extension mode, thus no velocity limits are initially implemented in the model.

Table 3.8: Torsion spring properties

Stiffness	0.1668 [kNm/deg]
Free position	43 [deg]
Rebound limit	50 [deg]
Rebound Stiffness	88 [kNm/deg]
Jounce limit	-4.59 [deg]
Rebound Stiffness	160 kNm/deg]
Arm length	0.3175 [m]

Table 3.9: Vertical damper properties

Velocity [m/s]	**Force [N]**	**Comments**
-1.27	-8896	Jounce mode
-0.254	-8896	
0	0	
0.254	2224	
1.27	2224	Extension mode

3.2.3 Powertrain

The NG-NRMM report provides a table with output torque for a single sprocket, at different vehicle speeds. No information about the transmission or steering is provided. To achieve the given output torque at the various speeds the `ChEngineSimpleMap` class is used together with the `ChAutomaticTransmissionSimpleMap` class and `ChSimpleTrackDriveline` class, resulting in a single continuous gear with sprocket output torque interpolated from the values specified in the engine class which can be seen in table 3.10.

Table 3.10: Sprocket torque at various sprocket speeds

Sprocket speed [rad/s]	0	0.88	2.51	5.01	7.52	10.02	20.04
Torque [Nm]	9519	9519	7140	3570	2379	1784	892

The vehicle reaches a top speed of around 30 mph, which is slower than the 40 mph stated in the NG-NRMM report [3]. One reason for the slower top speed is that the vehicle requires steering input to keep driving straight, which decreases the available torque when using the `ChSimpleTrackDriveline` class. This could likely be solved by tuning the PID driver that controls the vehicle.

Figure 3.5 shows the resulting torque at all speeds during a test run on rigid ground with high friction between the track and terrain, this requires stiffening the terrain and rubber track pads when using the SMC contact model. The valleys/dips in torque seen in the

figure are caused by oscillations in steering input from the PID driver. In the figure we can also see the idealized torque curve from the NG-NRMM report.

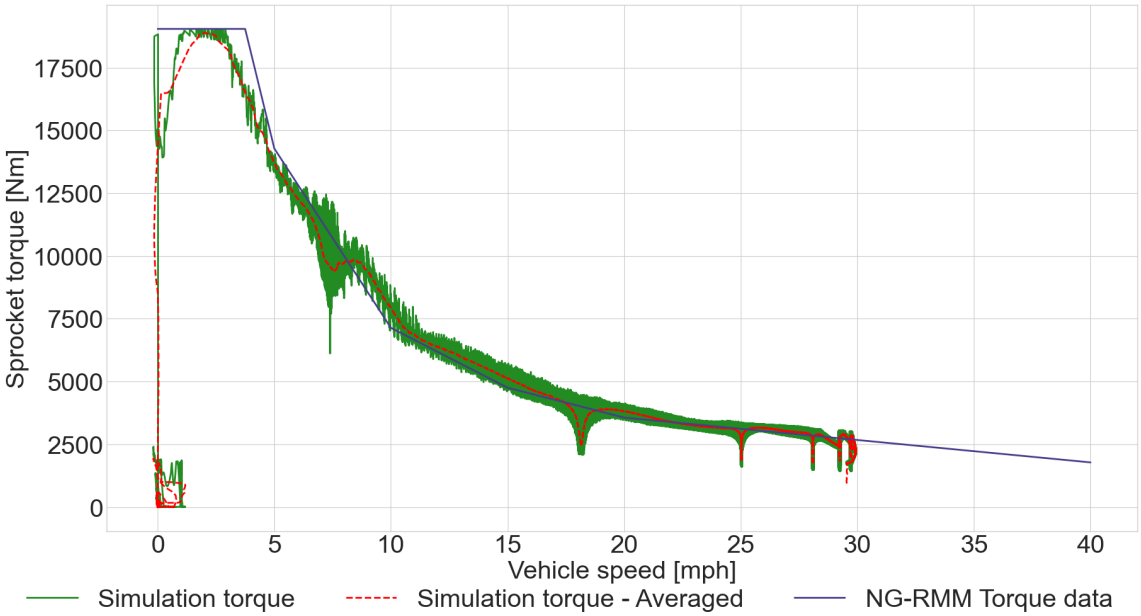

Figure 3.5: Sprocket torque at corresponding vehicle speed

3.2.4 Material properties

The NG-RMM [3] report specifies the radial stiffness for the idler wheel, sprocket and road wheel at 3.5E6 N/m, this value is used for the normal stiffness in the `ChContactMaterialData` in addition to the default values. The report states that the wheel rubber and track pad in series deflect 1 inch under the gross vehicle weight. To achieve this the Young modulus for the track shoe bottom pad was altered until the middle left road wheel center was deflected around 1 in when stationary, resulting in a value of 5E6 Pa. When measuring the deflection of the track shoe and wheel the ground stiffness was significantly increased, this results in some instability in the simulation (jittery/vibrating wheels and tracks), because of this a moving average using 100 data points was used to estimate the deflection. The moving average of the deflection and the target deflection can be seen in figure 3.6.

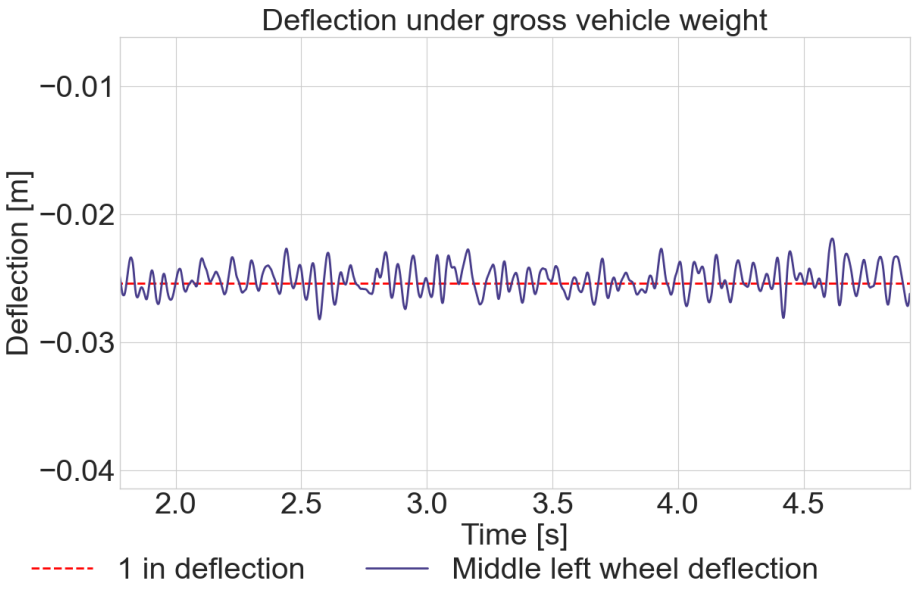

Figure 3.6: Moving average of middle left road wheel deflection using 100 data points

3.2.5 Track

The track is based on the default singlepin template. The weight of the track shoes is changed to 8.55 kg to match the data stated in [3]. The pitch and shape is kept as default. In [3] data is provided for initial track tension and track elasticity. The track elongation elasticity has not been taken into account, it can be altered via the material properties of the track shoes or the bushings between shoes, if bushings are used. The NG-RMM [3] report specifies an initial track tension of 10 kN. This is achieved by using the idler properties in table 3.11. The right side offset refers to the offset in the x-coordinate for the location on the chassis where the fixed point of the spring is attached. This is needed due to the left-right asymmetry of the road-wheel positions.

Finally, the bending stiffness of the track has been modeled using `ChLinkRSDA` joints with stiffness coefficient $k = 1000$ Nm/rad and damping coefficient $c = 10$ Nsm/rad. These values are inspired by [23] where the bending resistance of high-speed military vehicles using single-pin metal-link tracks with rubber-bushed joints are listed (as a footnote).

Table 3.11: Idler carrier properties

Mass	10 [kg]
Spring coefficient	1e6 [N/m]
Dampening coefficient	1.4e4 [Ns/m]
Free length	0.725 [m]
Force	1e4 [N]
Right side offset	0.075 [m]

4
Validating

4.1 Simulation Parameters

Before setting out to validate the model, some parameters related to the simulation have to be set. There is a large number of parameters that affect the numerical stability and general nature of the simulation, however only those identified as the most important were treated with scrutiny. These included: time step, solver and time stepper. Two potential solvers and two potential time steppers were identified. The solvers are Projected Successive Over-Relaxation (PSOR) [24] and Barzilai-Borwein (BB) [25]. The time steppers are Linearized Euler Implicit (Linear) [14] and Projected Euler Implicit (Projected) [15].

The choices of time steppers and solvers are based on the demos, the Chrono documentation – including the forum – and on heuristic test runs where the combinations of solvers and time steppers showed the ability to reliably handle simulations of tracked vehicles of various kinds, which cannot be said for all available solvers and time steppers.

To gain insight into how the size of the time step influences the simulation for the combinations of solvers and time steppers, a simulation with relatively large forces was run. The test entailed driving over a 12 inch (30.48 cm) half round obstacle at around 4.2 m/s. The steering and throttle input to the vehicle was managed by a PID controller that kept the vehicle driving in a straight line at the desired speed.

During the simulation the position of the vehicle's center of gravity (COG) was recorded to check that the desired speed and direction was kept for each run. The position of the road wheel was recorded because it was considered a highly dynamic output as it experiences a lot of force and displacement – and is governed by a spring-damper system – when driving over this type of obstacle. The simulation was then performed for a number of time step sizes using all four combinations of solvers and time steppers.

The purpose of these simulations is to observe how the trajectory of the road wheel is affected by lowering the time step, for each combination of solver and times stepper. The expected result is that the trajectory will converge to a stable trajectory as the time step is decreased. Then the combination of solver and stepper that show the fastest convergence is chosen.

This procedure was repeated for three different vehicle models. The vehicles will be referenced as Marder, M113 base and M113 alternative.

Marder: The Chrono demo model called Marder.

M113 Alternative: An intermediate step in the progress of modeling the M113 base model.

M113 Base: The M113 model described in chapter 3, using the default single-pin track, idler and engine.

4.1.1 Marder

The Marder was included because of its high structural complexity compared to the M113, due to having support rollers, more road wheels and more track shoes. This results in more contact forces and bodies to manage during the simulation. The idea was that an upper threshold of time step could be established by using a complex model.

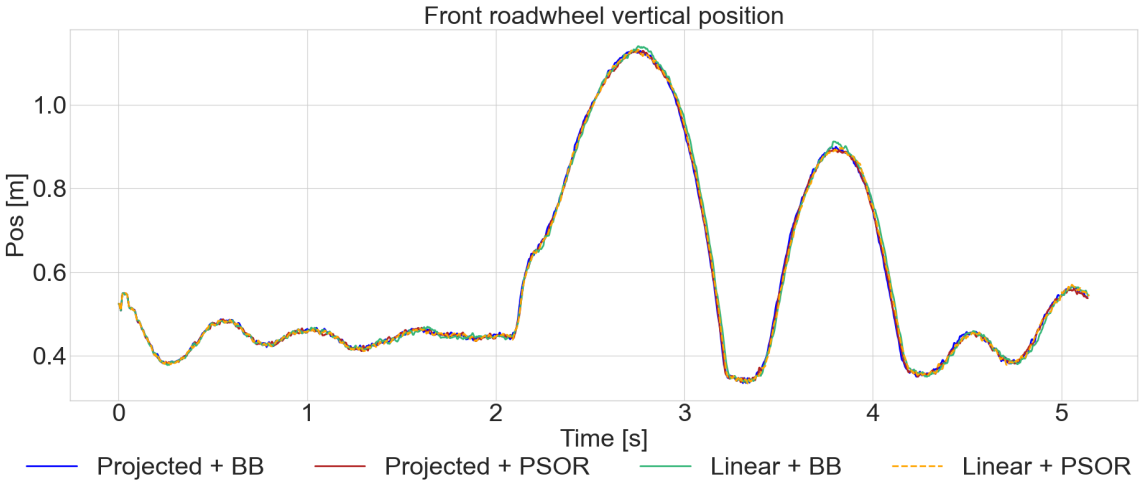

Figure 4.1: The front road-wheel vertical position of the Marder model for all solver-stepper combinations using the smallest evaluated time step of 5E-6 seconds.

Figure 4.2 shows the trajectories for all time steps and all time stepper solver combination. In figure 4.1 the trajectories from every combination using the smallest time step of 5E-6 seconds is shown in the same plot.

The result shows that the wheel trajectory is not noticeably affected by the size of the time step, solver or time stepper. The fast convergence of the wheel trajectory for every combination of time stepper and solver can partly be explained by the fact that the torsion bar suspension has no jounce stiffness. This means that the front wheel can travel vertically relatively freely when it hits the half round obstacle and later the ground, without being exposed to particularly stiff conditions.

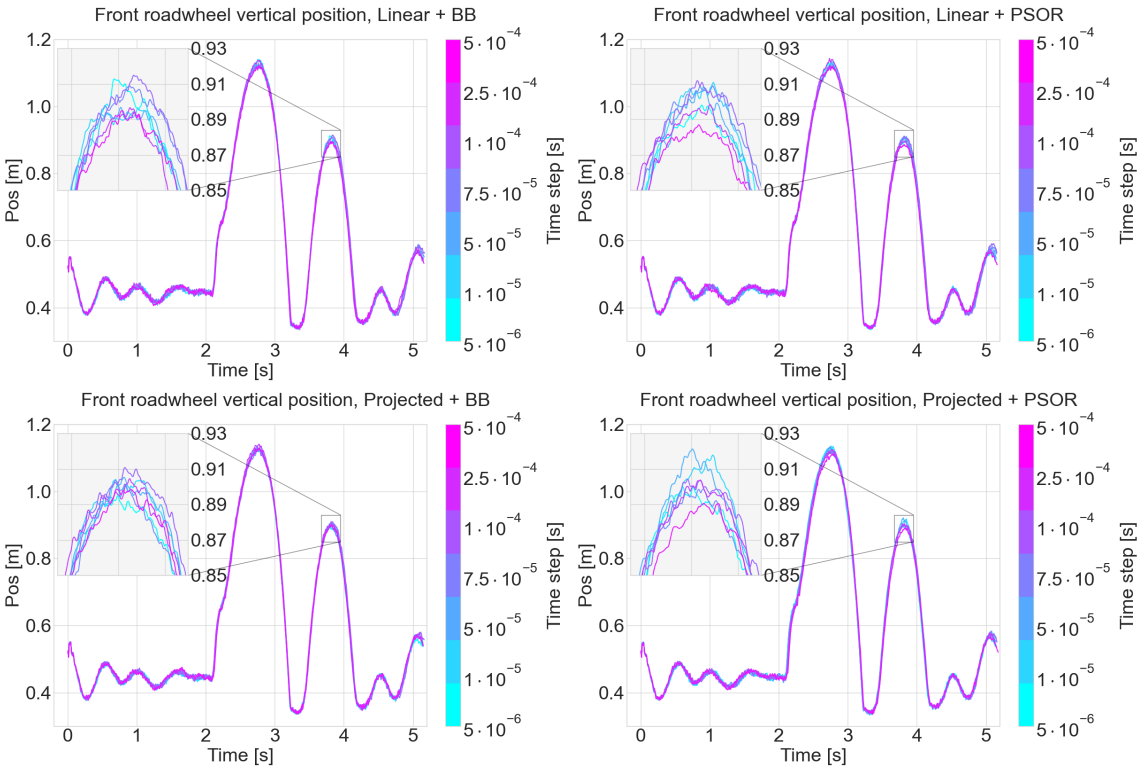

Figure 4.2: The trajectories of the front road-wheel vertical position of the Marder model for all solver-stepper combinations and time steps.

4.1.2 M113 Alternative

The M113 alternative model is an unfinished intermediate step in the process of converting the M113 Chrono demo model to the base model described in chapter 3. The most important differences are a 10 % lower weight, three times as stiff torsion bar suspension without any jounce stiffness and significantly lower damping coefficient on the first and last wheel.

Figure 4.4 shows the trajectories for all time steps and all time stepper solver combinations. In figure 4.3 the trajectories from each combination using the smallest time step of 5E-6 seconds is shown in the same plot.

In contrast to the Marder, the M113 alternative model shows large deviations between the different time steps. However all time stepper and solver combinations except projected with PSOR eventually converge to roughly the same trajectory, as can be seen in figure 4.3. A big reason for the large differences between time steps is likely because of the decreased numerical stability due to stiffer suspension and lower weight.

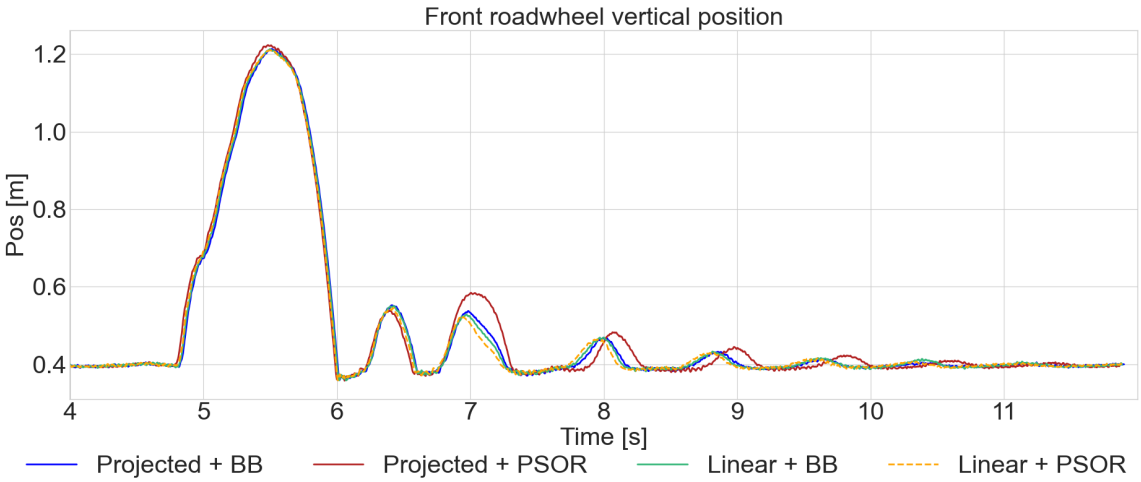

Figure 4.3: The front road-wheel vertical position of the M113 alternative model for all solver-stepper combinations using the smallest evaluated time step of 5E-6 secounds.

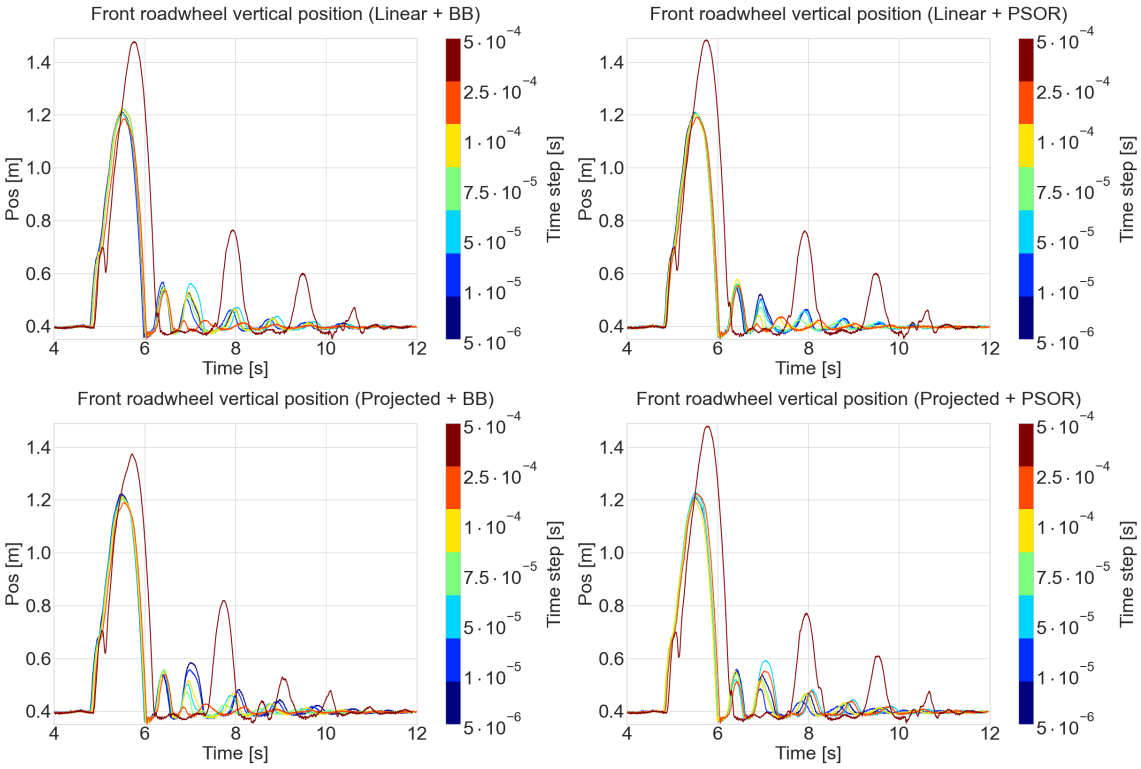

Figure 4.4: The trajectories of the front road-wheel vertical position of the M113 alternative model for all solver-stepper combinations and time steps.

4.1.3 M113 Base

The M113 Base model is identical to the model described in chapter 3, except for the idler, engine and Young modulus for the track shoe bottom pad which are the default values. The biggest differences from the Marder and M113 alternative models are the added jounce and rebound stiffness on the torsion bar suspension and the non constant damper coefficient.

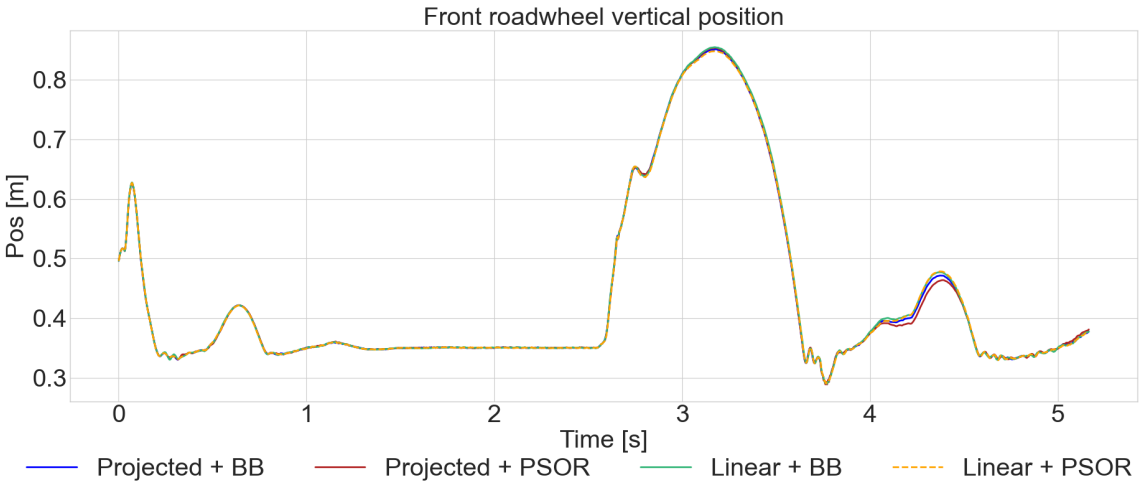

Figure 4.5: The front road-wheel vertical position on the m113 base model for all solver-stepper combinations using the smallest evaluated time step of 5E-6 secounds

Figure 4.6 shows the trajectories for all time steps and all time stepper and solver combinations. In figure 4.5 the trajectories from every combination using the smallest time step of 5E-6 seconds is shown in the same plot.

Compared to the M113 alternative model the wheel trajectories display minimal change when the size of the time step is decreased, however it is more sensitive to the time step than the Marder model. Looking at the trajectories around the 4 second mark we can see that all combinations of time steppers and solvers – regardless of time step size – lead to the road wheel oscillating at impact with the ground, just before briefly penetrating the ground. This is due to the aforementioned jounce stiffness stopping the vertical travel of the wheel. This effect is mostly using a very high damping coefficient in the vertical dampers when they reach high speeds, as this prevents the wheel from entering the stiff jounce territories of the torsion bar suspension.

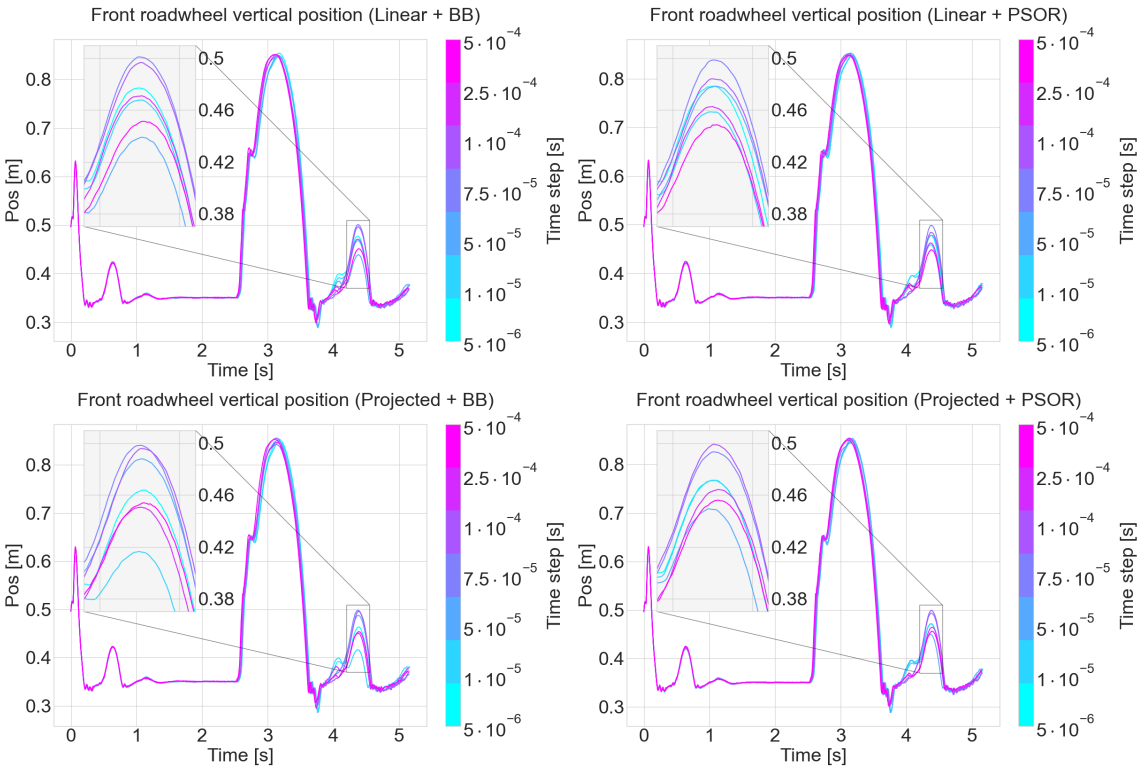

Figure 4.6: The trajectories of the front road-wheel vertical position of the M113 base model for all solver-stepper combinations and time steps.

4.1.4 Resulting Parameters

Judging by the results above all combinations of time steppers and solvers except projected with PSOR produce comparable results. Thus the final choice of time stepper and solver is instead made based on the Chrono demos of tracked vehicle simulations, where the most common combination was the linearized Euler implicit time stepper and the Barzilai-Borwein solver [26] [27].

The size of the time step was chosen as 1E-4 seconds, as this produced trajectories close to the trajectory with the smallest time step when using the Linear + BB combination. This is especially evident when looking at the wheel trajectory of the M113 alternative model in figure 4.4.

In addition to time stepper, solver and time step, the parameters displayed in table 4.1 are specified. The parameters are the same as in the tracked vehicle demos of [26] [27], with the exception of max iterations which was increased to 200. This was done because any lower value resulted in a visual violation of the revolute joint constraints attaching the torsion arms to the chassis when the wheels hit the half round obstacle.

Table 4.1: Simulation parameters

Name	Value
Contact Model	SMC/DAE
Force Model	Hertz
Time Stepper	Euler Implicit Linear
Time step	1E-4
Solver	Barzilai-Borwein (BB)
Max Iterations	200
Omega	0.8
Sharpness Lambda	1.0
Max penetration recovery speed	1.5
Min bounce speed	2

4.2 Validating the M113 Base Model

To be able to draw any meaningful conclusion from vehicle simulations, the behavior of the model needs to correspond to the behavior of the real vehicle one sought out to model. Based on the NG-NRMM report [3] and other validation studies [9], it is assumed that the Chrono software is capable of simulating the M113. The second assumption that will be made is that the simulation parameters specified in section 4.1 are appropriate for the relevant simulations. Thus the only aspect left to validate is the model itself. Ideally this is achieved by comparing the simulated data to data collected from the corresponding vehicle and finding the cause of any deviation between the two. Unfortunately this is not an option and instead the M113 base model will be compared to the results presented in the NG-NRMM report where the NRMM software and the vendors (table 2.1) have simulated the M113 in various events.

Initially the idea was to only compare the model to the results presented by the Chrono developer team, as this would eliminate differences in results caused by differing software. This however turned out to be unfruitful as there is still too much uncertainty in how the model is implemented and how the simulation is set up. There is also nothing in the NG-NRMM report that explicitly states that Chrono's results are more accurate than any other result with the same maturity level. Thus it was decided to compare the model to the result of all vendors producing the highest maturity level for each event.

Not all events presented in the NG-NRMM report were used for validation. In fact only a handful were chosen based on how relevant they were from the perspective of mobility over rigid terrain. If nothing else is explicitly stated, the ground and obstacle used have a friction coefficient of 0.8 as stated in the NG-NRMM report and a Young modulus of 20 GPa to emulate concrete [28].

4.2.1 Half Round Obstacle Ride Limiting Speeds

This event is based on the Test Operations Procedure (TOP) 1-1-014 [29]. The procedure entails driving over a half round rigid obstacle with increasing speed until a peak vertical acceleration of 2.5 g is measured at the driver's position. This was achieved by initializing the vehicle with the target speed at a distance equal to three times the target speed (in m/s) before the obstacle. The vehicle was then kept driving in a straight line at a constant speed using a PID controller. Thus the vehicle hit the obstacle after roughly three seconds, giving it enough time to stabilize. This was repeated for a number of target speeds over half-round obstacles of height 4, 6, 8, 10 and 12 inches. The limiting speed for each obstacle was estimated using linear interpolation, i.e. the speed where the lines intersect the red 2.5g limit in figure 4.8 and 4.7 .

Figure 4.8 shows the peak vertical acceleration measured at the driver's position at different speeds for half round obstacles of height 8, 10 and 12 inches and figure 4.7 show the same for half round obstacles of height 4 and 6 inches. The displayed values are the absolute peaks from the unfiltered acceleration data.

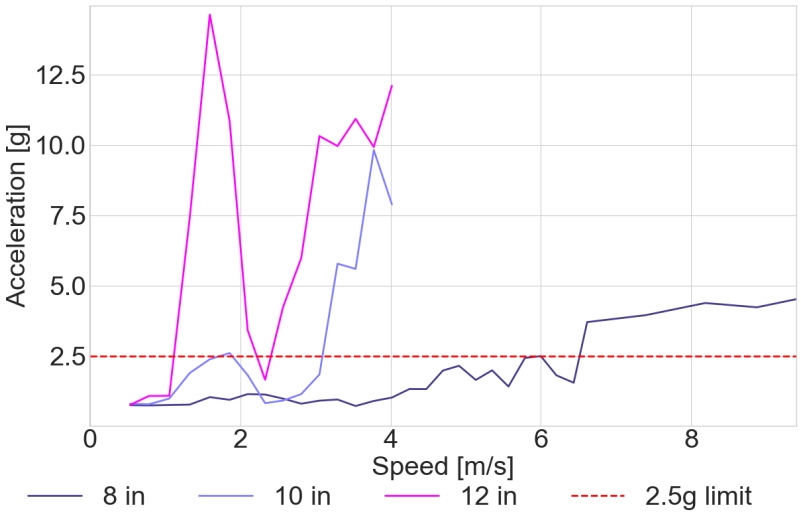

Figure 4.7: Peak vertical acceleration measured at the driver's position at different speeds for half round obstacles of height 8, 10 and 12 inches.

The NG-NRMM report provides no empirical data for this test, meaning the highest achievable maturity rating was 4, see table 2.2. Vendors B, C, D and NRMM have all acquired a maturity level of 4, hence their values will be the benchmark. Figure 4.9, 4.10, 4.11, 4.12 and 4.13 shows the acquired values compared to the values presented in the NG-NRMM report.

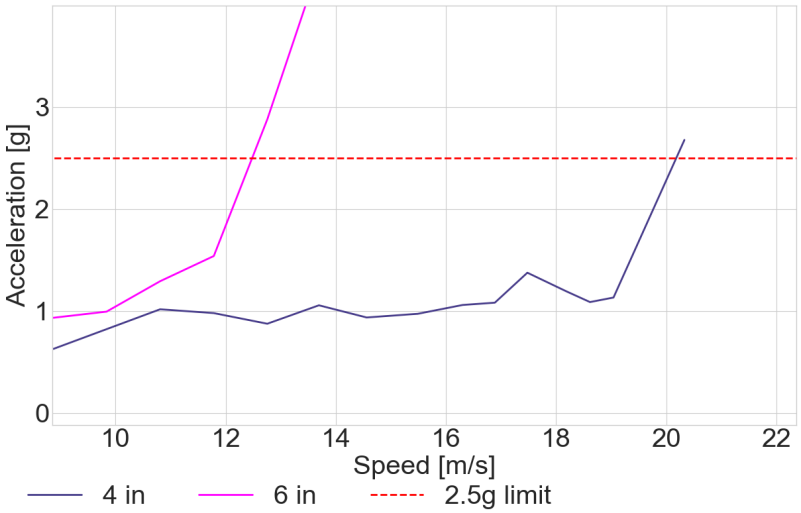

Figure 4.8: Peak vertical acceleration measured at the driver's position at different speeds for half round obstacles of height 4 and 6 inches.

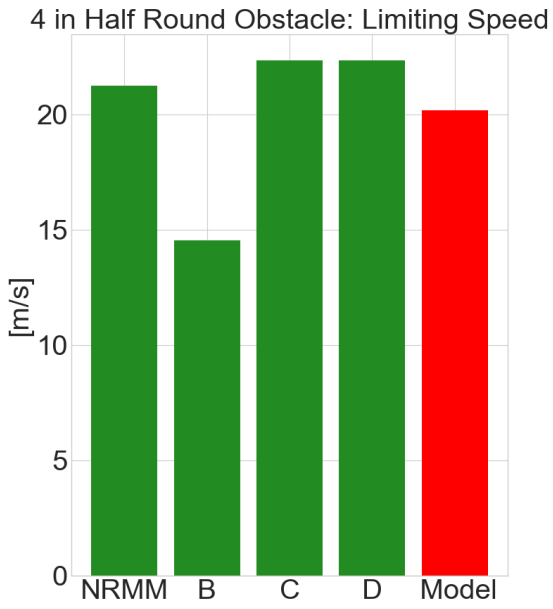

Figure 4.9: The limiting speeds over a 4 inch half round obstacle found by NRMM and the vendors B, C and D as presented in the NG-NRMM report and the acquired result from the simulation in red.

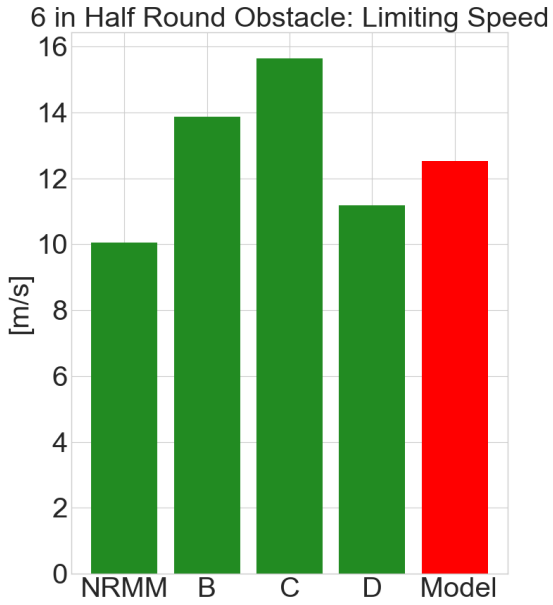

Figure 4.10: The limiting speeds over a 6 inch half-round obstacle found by NRMM and the vendors B, C and D as presented in the NG-NRMM report and the acquired result from the simulation in red.

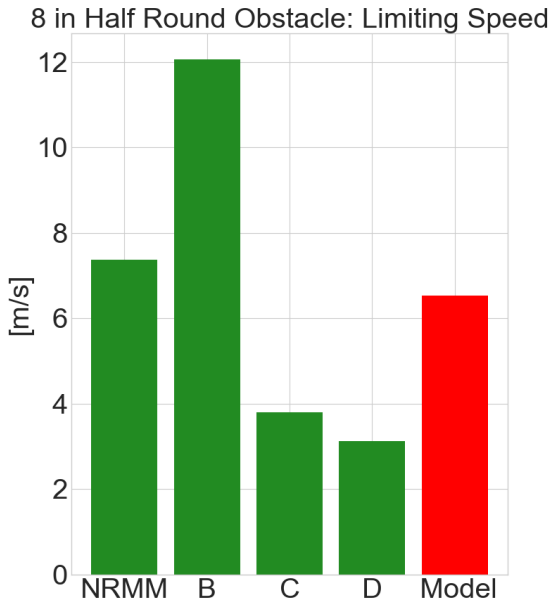

Figure 4.11: The limiting speeds over a 8 inch half round obstacle found by NRMM and the vendors B, C and D as presented in the NG-NRMM report and the acquired result from the simulation in red.

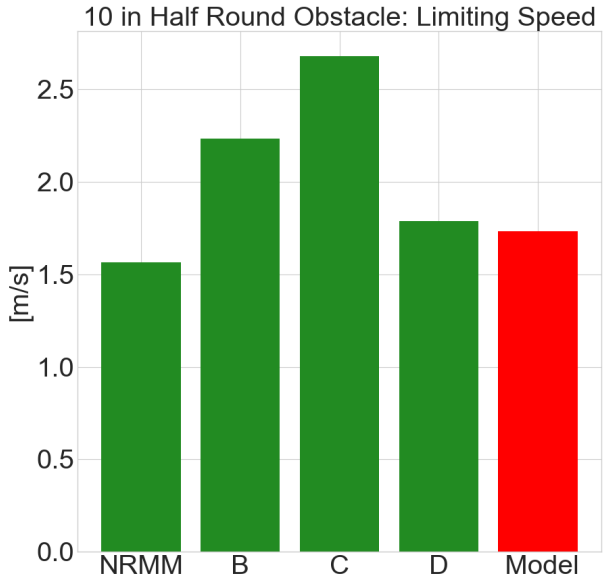

Figure 4.12: The limiting speeds over a 10 inch half round obstacle found by NRMM and the vendors B, C and D as presented in the NG-NRMM report and the acquired result from the simulation in red.

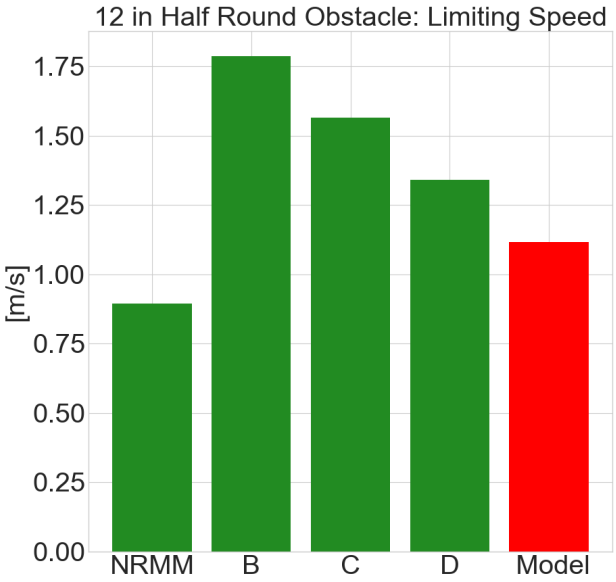

Figure 4.13: The limiting speeds over a 12 inch half round obstacle found by NRMM and the vendors B, C and D as presented in the NG-NRMM report and the acquired result from the simulation in red.

The acquired results are comparable to the results of NRMM and the vendors, for all obstacles it falls between the highest and lowest. The acceleration over both the 10 in and 12 in obstacle show a pattern of first increasing then decreasing before increasing again. This first sharp increase is due to the idler hitting the obstacle hard at certain low speeds, when the speed then is slightly increased the idler clears the obstacle. The subsequent increase in acceleration is due to the spring hitting its jounce stop with increasingly higher speed. For the 4, 6 and 8 inch obstacle the vertical travel speed for the front wheel (equipped with a vertical damper) greatly exceeds 1.27 m/s.

4.2.2 Step Obstacle

This event is based on the TOP 2-2-611 standard [30]. Here the goal is to find the maximum height of a vertical step obstacle which the vehicle can traverse. This was achieved by first initializing the vehicle as close as possible to the obstacle, then after two seconds when the vehicle had stabilized, the throttle was set to its maximum. If the vehicle could climb the obstacle within 20 seconds the step height was increased by 2 cm and the procedure was repeated until the height limit was found. The TOP standard specifies that the step should have ”replaceable timbers with a squared edge on the top...”, thus the Young modulus of the step was lowered to 10 GPa, consistent with the values presented in [31].

Figure 4.14 shows the trajectory of the center of gravity in the x-z plane of the vehicle successfully climbing a step obstacle of 78 cm and failing to climb a step obstacle of 80 cm.

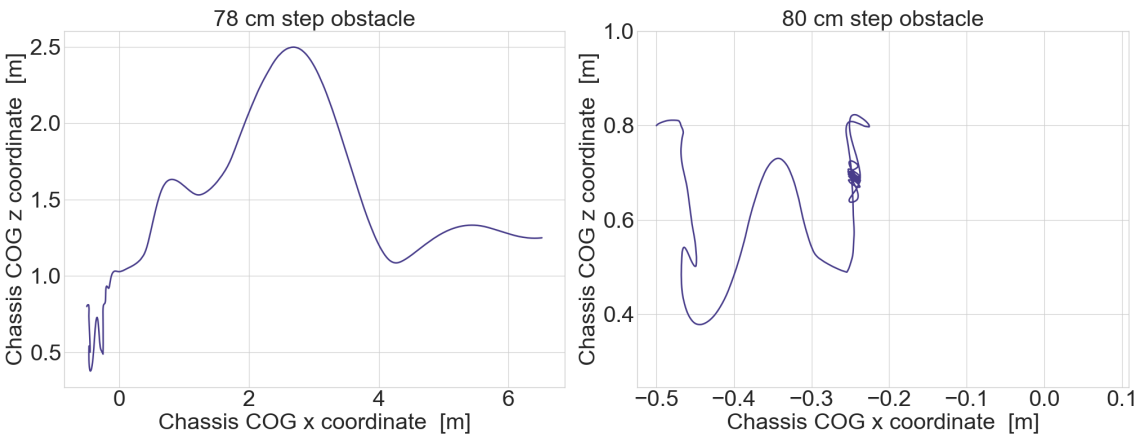

Figure 4.14: The trajectory of the center of gravity in the x-z plane of the vehicle successfully climbing a step obstacle of 78 cm and failing to climb a step obstacle of 80 cm.

The NG-NRMM report provides empirical data for this event, meaning the highest achievable maturity rating was 5. Vendors A, B, C and D have all acquired a maturity level of 5 and their results together with the empirical result in blue and the acquired result from the simulation can be seen in figure 4.15.

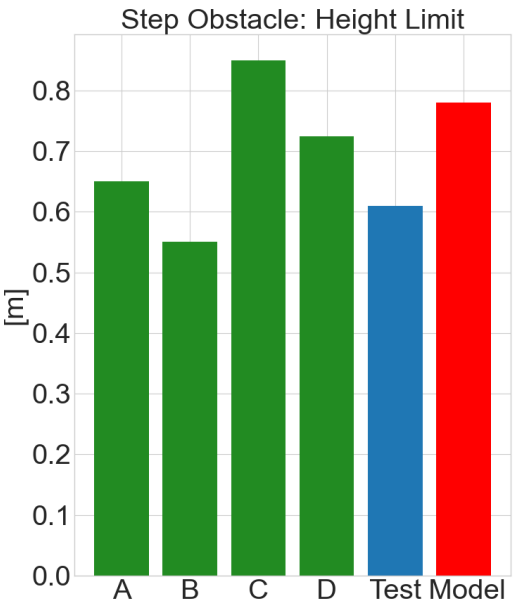

Figure 4.15: The step height limit found by vendors A, B, C and D as well as the empirical test data presented in the NG-NRMM report and the acquired result from the simulation.

The acquired results are comparable to the results of the vendors and higher than the empirical data. The visual rendering of the simulation shows that the limiting factor was friction, which is also indicated by the non-stationary trajectory of the failed attempt in figure 4.14.

4.2.3 Gap Obstacle

This event is based on the TOP 2-2-611 standard [30]. Here the goal is to find the maximum gap that a vehicle can span. The TOP standard specifies that the vehicle should be driven at minimal speed over the gap, stopping frequently to minimize the influence of momentum. For the sake of repeatability the simulation simply used a straight line PID driver with a target speed of 0.36 m/s. The gap was increased by 3 cm at a time until the vehicle could not traverse the gap.

Figure 4.16 shows the trajectory of the center of gravity in the x-z plane of the vehicle successfully traversing a gap of 208 cm and failing to traverse a gap of 211 cm.

The NG-NRMM report provides empirical data for this event, meaning the highest achievable maturity rating was 5. Vendors A, B, C and D have all acquired a maturity level of 5 and their results together with the empirical result in blue and the acquired result from the simulation can be seen in figure 4.17.

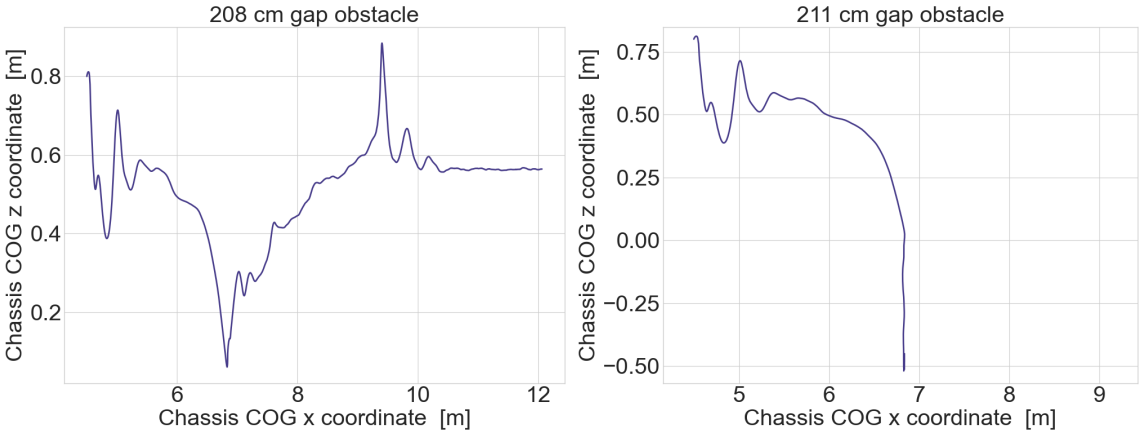

Figure 4.16: The trajectory of the center of gravity in the x-z plane of the vehicle successfully traversing a gap of 208 cm and failing to traverse a gap of 211 cm.

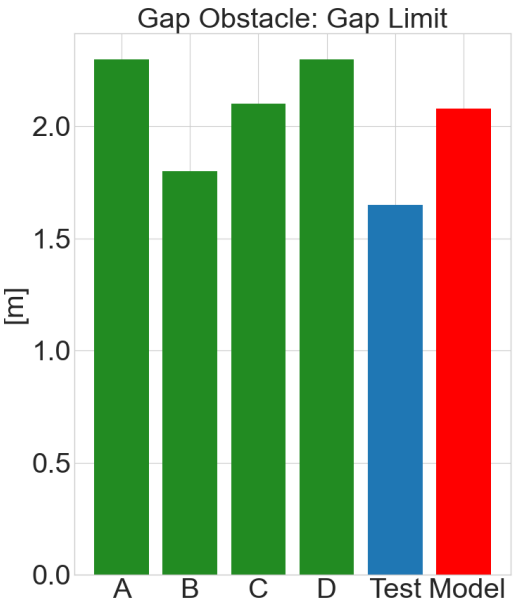

Figure 4.17: The gap crossing limit found by vendors A, B, C and D as well as the empirical test data presented in the NG-NRMM report and the acquired result from the simulation.

The acquired results are comparable to the results of the vendors and higher than the empirical result. One reason for the higher value could be a difference in the operation of the vehicle, the PID driver used in the simulation tried to achieve a constant speed while the real vehicle would have stopped frequently to stabilize, if the procedures of TOP 2-2-611 was followed.

4.2.4 Trapezoidal Obstacles

In this event 12 different trapezoidal barrier obstacles with a height of 30in are created from a combination of the following parameter values: table top lengths of 6in, 30in, 140in and slope angles of 16 deg, 26 deg, 38 deg and 68 deg. These 12 barriers were then inverted to act as trapezoidal ditches.

The NG-NRMM report provides no empirical data for this test, meaning the highest achievable maturity rating was 4. Vendors A, B, C, D and NRMM have all acquired a maturity level of 4, hence their values will be the benchmark. The result is presented either as a successful or failed attempt to traverse the obstacle. The result from the barriers is shown in figure 4.18 and the result from the ditches are shown in figure 4.19.

	1	2	3	4	5	6	7	8	9	10	11	12
NRMM	S	S	S	F	S	S	S	S	S	S	S	S
A	S	S	S	S	S	S	S	S	S	S	S	S
B	S	S	S	F	S	S	S	F	S	S	S	F
C	S	S	S	S	S	S	S	S	S	S	S	S
D	S	S	S	S	S	S	S	S	S	S	S	S
Model	S	S	S	S	S	S	S	S	S	S	S	S

Figure 4.18: Trapezoidal barrier crossing result, the green S denotes a successful crossing and the red F denotes a failed crossing

The model manages to navigate all obstacles except the 140 inches long ditch with 68 degrees approach angle, where it gets stuck with the sprocket against the wall and the idler still outside the ditch. This is a comparable result to the participating vendors.

	1	2	3	4	5	6	7	8	9	10	11	12
NRMM	S	S	S	S	S	S	S	S	S	S	S	F
A	S	S	S	S	S	S	S	S	S	S	S	S
B	S	S	S	S	S	S	S	S	S	S	S	S
C	S	S	S	S	S	S	S	S	S	S	S	F
D	S	S	S	S	S	S	S	S	S	S	S	F
Model	S	S	S	S	S	S	S	S	S	S	S	F

Figure 4.19: Trapezoidal ditch crossing result, the green S denotes a successful crossing and the red F denotes a failed crossing

4.3 Discussion of Validation

Beginning with the simulation parameters, we can conclude the method used is arguably not very effective since it did not give an unambiguous answer to the problem of choosing the simulation parameters. The method did however highlight the fact that different models require different parameters to be successfully simulated. Ideally a more thorough investigation into why the M113 alternative was unstable would be carried out. Moreover it is unclear how the rest of the simulation parameters in table 4.1 affect the simulation. Nonetheless, for now it is considered sufficient to acknowledge that the M113 base model seems stable with the chosen parameters.

The simulations in section 4.2 also called attention to the problem of not implementing any velocity limiting damping on the vertical dampers attached to the front and rear road wheels. This is further underlined in the simulations over the half round obstacle of size 4, 6 and 8 inches, where the vertical speeds of the dampers exceeded the 1.27 m/s stated in the NG-NRMM report. However the report does not specify how the dampers should behave at high speed; additional simulations suggest that this is the most important factor for determining the limiting speed over obstacles at high speed. Therefore it is decided to avoid further simulations where the road wheels are exposed to vertical velocities that exceed 1.27 m/s.

The remaining validation simulations were all carried out at a slow enough speed for the vehicle to exhibit well defined behavior. The results are also reasonable when compared to those displayed in the NG-NRMM report. Unfortunately the report only discloses the bare minimum about the results, it is not mentioned in what manner the real vehicle succeeded or failed in the various events. For example if the hull of the chassis ever interferes or if the step climb height is limited by friction or power.

Based on the visual representations of the simulations, the model exhibits much more slip between the track and ground than seen in, for example the step climbing procedures of [32] [33] [34]. This is partly due to differences in driving technique, where the simulation uses full throttle to climb the steps. It is also likely because of too low friction between the track and ground, caused partly by the low Young modulus of the track shoe bottom pad.

Both the obstacle and track shoe bottom pad have the same default Poisson's ratios. By using equation (2.19) we see that the change in Young modulus from the default 0.01 GPa to 0.005 GPa causes the effective shear modulus between the track shoe and the step to decrease by about 50%. The tangential force coefficient (2.18) decreases by the same factor which potentially results in half the frictional force according to equation (2.16). The actual decrease in frictional force depends – among other factors – upon the interpenetration between the two bodies which in turn depends on the bodies' Young modulus. The point being that the change in Young modulus has the potential to significantly affect the frictional force.

That being said, lowering the Young modulus to achieve a stable simulation is a common practice when using a DAE/SMC approach to multibody dynamics [17] [8] and additional simulations with the default – and even higher – track shoe Young modulus also exhibit lower friction than expected. This suggests that the friction behavior is inherent to the current simulation setup using this particular contact model, force model and simulation parameters. This could perhaps be remedied by relying on explicitly defined material stiffness coefficients (2.17) (2.18) rather than deriving them from material properties.

Another approach is using the NSC contact model which does not rely on interpenetration to calculate frictional and normal forces as it treats all bodies as completely rigid. When performing step climb simulations with the NSC contact model the simulated vehicle shows no track slip and behaves closer to the vehicles in [32] [33] [34] resulting in a step climb limit closer to the experimental result in section 4.2.2. It might initially be surprising that the high friction simulation results in a lower step climb limit, the reason is that the low friction SMC simulation allows the spinning sprocket to catch the top of the step edge and transfer some of its angular momentum to the effort of pulling the vehicle over the step, this phenomenon is also explained by the Chrono software team in [22].

Despite the many uncertainties in how the simulations and experiments were carried out and how the models were implemented, the model produces similar results to those displayed in the NG-NRMM report with the highest maturity levels. Although, there is still a lot of variation between these results and not always experimental data to compare with. The model is far from ready to be used for quantitative results but does potentially exhibit reasonable behavior in certain conditions. For the purpose of the thesis, let us temporarily assume that the model and the simulation setup is adequate for exploring how the geometry of the track-assembly affects the mobility over rigid terrain at low speeds, albeit at a perhaps lower friction than envisioned.

5
Evaluating

With the M113 modeled and validated, we are ready to evaluate different geometries of the track-assembly. Specifically, varying the horizontal and vertical positions of the idler (idler x and idler z in figure 5.1) and sprocket (sprocket x and sprocket z in figure 5.1). The performance of the resulting track-assembly configurations will be evaluated in various events featuring different obstacles.

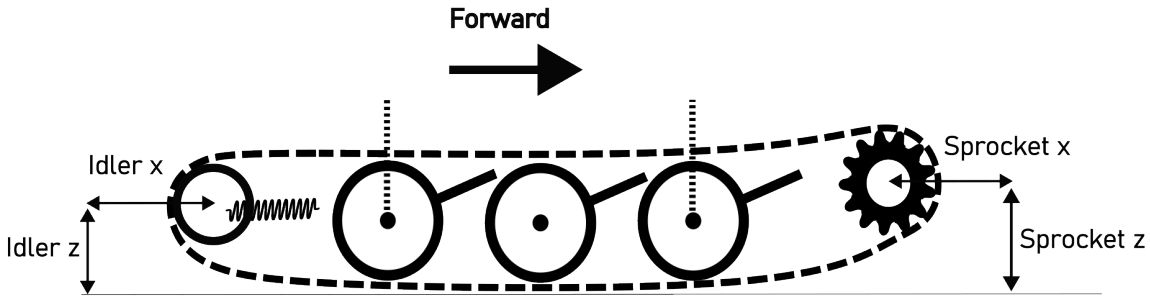

Figure 5.1: Schematic image of the parameters that will be varied in the evaluation.

5.1 Evaluation model

Before altering the now validated model to evaluate different track-assembly configurations, a few changes had to be made to the model. First the left side was set to mirror the right side, i.e. the left-right asymmetry from the base model is discarded. This change was done to get the same distance between the sprocket and front road wheel on both sides, enabling the same sprocket positions for both sides, and to simplify the algorithms that rectify conflicting parameters. The second change was to stiffen the vertical dampers

to 1e5 Ns/m when their vertical speed exceeds 1.27 m/s. This is done to keep the velocity within the modeled range and has minimal effect during low vehicle speeds.

Ideally no changes would be done to the validated model, as any changes risk nullifying the previous validation. However, the executed changes were deemed necessary for generalizing the vehicle model and thus allowing for a wider range of configurations with as little influence on the remaining parameters as possible. Furthermore, as is indicated by the results, the impact that these changes have on the vehicle behavior are likely insignificant in relation to the goal of evaluating how the geometry of the track-assembly affects the vehicle's mobility.

To run the evaluation simulations described below the model needs to be dynamic in the sense that parameters can be changed during runtime. Section 3.1.2 explains how the parameters are exposed for change but does not deal with interdependence between parameters. There is no one fits all solution to this, as all pairs of parameters have different relationships. For this thesis, only the position of the idler and sprocket are varied close to their default positions. This simplifies things greatly and we only need to consider a couple of conflicting parameters.

The first potential conflicting parameter is the number of track shoes, as the track length needs to change between different track-assembly configurations. This problem was solved by using a modified copy of the `ChTrackAssemblySinglePin.Assemble` function to calculate the required number of track shoes for a successful assembly. The second conflicting parameter is the location of the closest road wheel, as placing them too close risks collision or interpenetration. This is solved by simply checking that the distance between the centers of the relevant wheels is greater than a certain threshold.

Lastly there is the problem of track tension. The way the track is tensioned in Chrono is by the idler exerting a force on the track, due to being attached to the chassis with a spring. This setup allows the idler to move in the spring's direction, which in this case is the horizontal direction. Thus if the length of the track is changed, the force from the idler and the track tension changes. To achieve the same track tension for every track-assembly configuration, the idler is initially spring-less and is set to simply exert the 20 kN of force required to get an initial track tension of 10 kN. When the vehicle simulation reaches an equilibrium, the horizontal position of the idler is recorded for an amount of time while tensioning the track with the required force. The idler is then "locked in place" at the average horizontal position recorded during equilibrium, by reintroducing the spring.

5.2 Sprocket Position

Event 1: The sprocket position was varied and the height limit for a step obstacle was determined for each configuration. The vertical and horizontal sprocket coordinates were varied from +20 cm to -20 cm from the default position with 5 cm intervals. The positions where the sprocket intersected with the front road wheel were discarded resulting in

a total of 69 evaluated configurations .

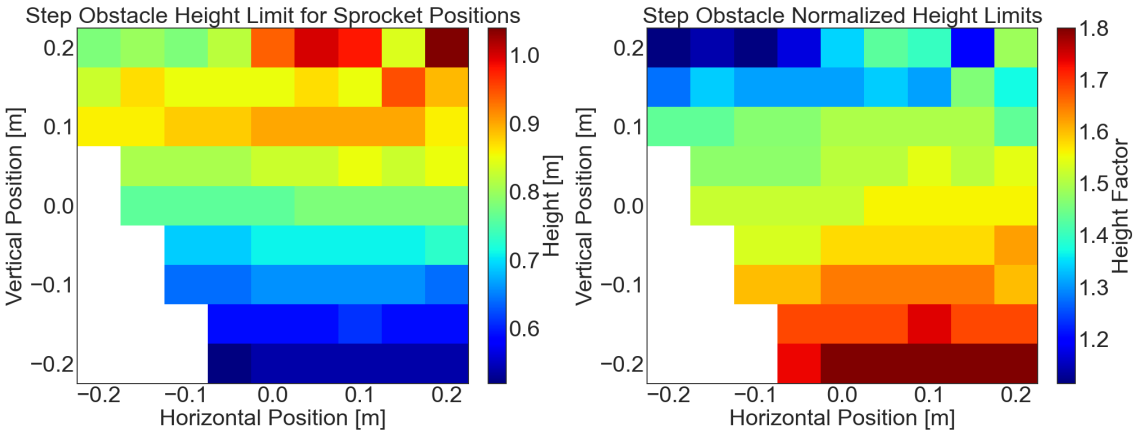

Figure 5.2: The resulting obstacle height limit represented by a color from the color bar for each sprocket position (left). The obstacle height limit, divided by the height from the ground to the sprocket, for every sprocket position (right).

In figure 5.2 the resulting obstacle height limit for each sprocket position can be seen in the left image. The right image in the same figure shows the height limit divided by the height from the ground of the sprocket, a normalized version of the left image. The default position, with vertical and horizontal position of 0, resulted in a height limit of 0.76 [m]. This is comparable to the 0.78 [m] found when validating in section 4.2.2, where the difference is due to the changes mentioned in the beginning of the chapter. In figure 5.3 the same data is shown but with the limiting height as a function of the vertical (left) and horizontal (right) position separately.

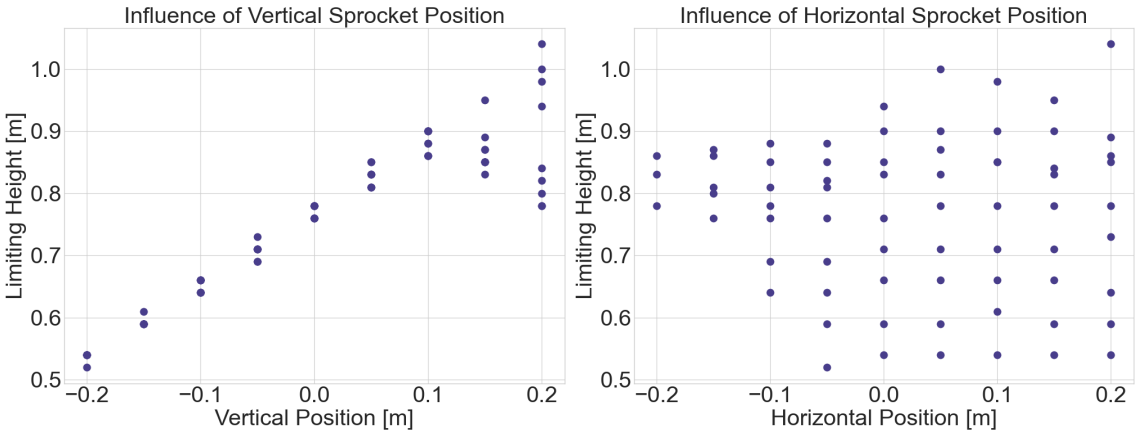

Figure 5.3: The obstacle height limit as a function of the vertical (left) and horizontal (right) sprocket position.

As expected a higher positioned sprocket resulted in a higher step limit. It was considered that a negative horizontal position could lead to higher step climb limit, because a steeper

track angle would lead to a larger contact area between the track and the obstacle, generating more friction and thus a higher climb limit. However looking at the right image of figure 5.3 one can conclude that no clear correlation between the horizontal sprocket position and climb height limit is shown.

The upper left corner of both images in figure 5.2 is of interest as it breaks the pattern where a higher sprocket position results in a higher climb limit. This can also be seen in the downward going branch of figure 5.3. After rendering a visualization of these simulations a plausible explanation was found. In these cases the sprocket manages to climb the obstacle just fine, however the front road-wheel does not. This is thought to be because the force from the obstacle on the track and road-wheel is aligned with the suspension arm. This yields a relatively small torque on the suspension arm – because of the small leverage – and the road-wheel is not vertically lifted. See figure 5.4 for a visual representation.

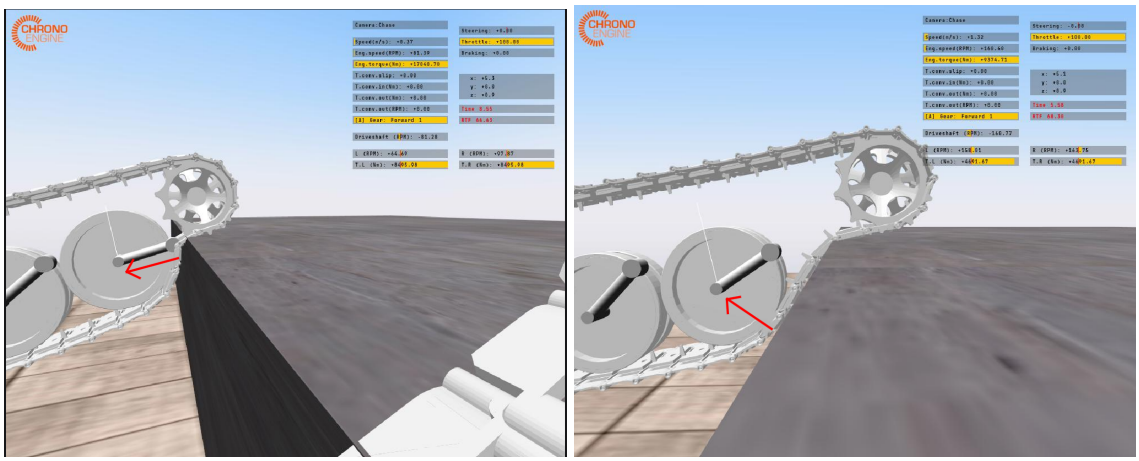

Figure 5.4: Two snapshots from the step climb simulation. The left image shows the vehicle with the sprocket positioned 20 cm backwards and upwards. The right image shows the vehicle configured with the sprocket 20 cm forwards and upwards. The red arrows were added post simulation to suggest the direction of the force from the obstacle to the road wheels.

Because of the track tensioning technique explained in section 5.1, every configuration has a slightly different initial track tension (ITT) and horizontal idler position. To make sure that this does not influence the results in a severe way these properties are recorded for every simulation. In figure 5.5 the track tension and idler position is shown. The left image displays the average track-shoe tension of all track-shoes during the equilibrium period in the beginning of the simulation, the result is slightly lower than the target ITT of 10 kN because of the lower tension in the track shoes under the road-wheels [35]. The right image show the horizontal coordinate of the idler, measured from the original sprocket position.

The differences in ITT are considered negligible, and there is no apparent correlation between the ITT in figure 5.5 and the height limit in figure 5.2. Regarding the idler position,

there are some relatively large deviations of idler position although no apparent correlation between the idler coordinate in figure 5.5 and the height limit in figure 5.2 is found. The effect of the idler position will be further investigated in section 5.3.

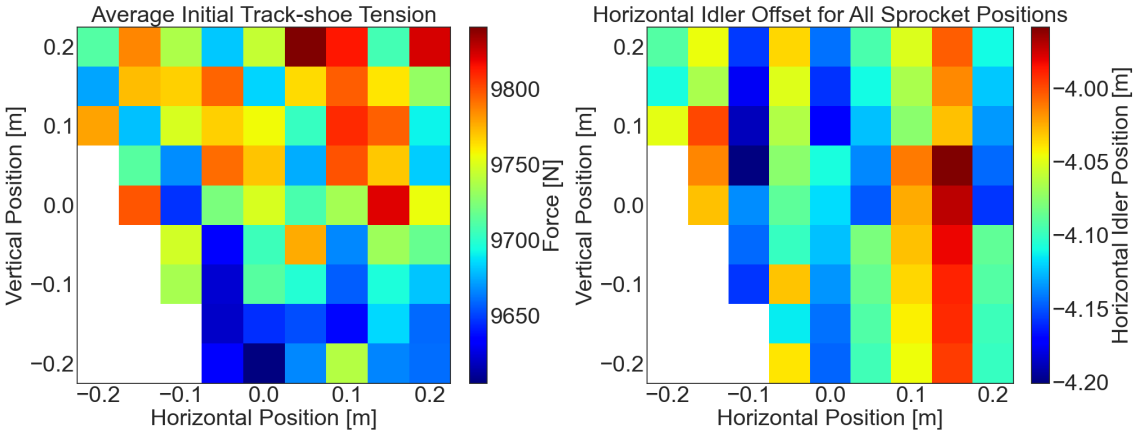

Figure 5.5: The average initial track shoe tension (left) and the horizontal idler position (right) for every evaluated sprocket position.

5.3 Idler Position

Due to the track tensioning technique explained in section 5.1, the horizontal idler position is determined by the length of the track. This means that the actual idler location might differ from the input to the `SetIdlerLocation` function. Another consequence of the track tensioning technique is that multiple different idler location inputs can end up with the same final idler location, because the possible horizontal locations are actually discrete for a given vertical position due to the track consisting of discrete track shoes. This means that very similar track configurations were evaluated multiple times. When multiple similar configurations have different results, the background color around that point is set to the average result. This applies to the right plot of figure 5.6, 5.8, 5.9 and 5.10.

Event 2: The idler position was varied and the height limit for a step obstacle was determined for each configuration. The vertical and horizontal idler coordinates were varied from +20 cm to -20 cm from the default position with 5 cm intervals.

The left image of Figure 5.6 shows the resulting obstacle height limits for each idler position input as an offset from the default position. The right image shows the maximum obstacle height as a function of the actual idler position, indicated by the black circles. The default position resulted as expected in a height limit of 0.76 [m], as in event 1. In figure 5.7 the same data is shown but with the limiting height as a function of the vertical (left) and horizontal (right) position separately.

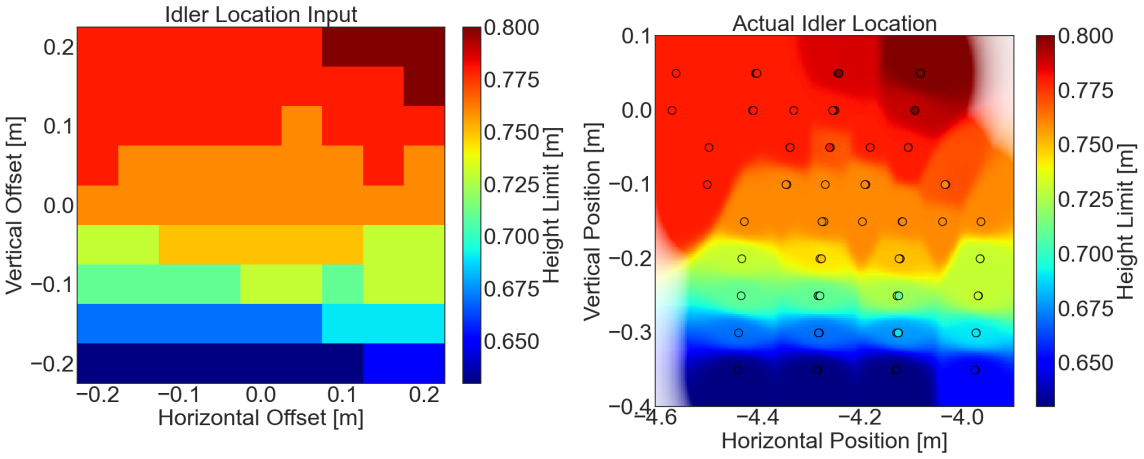

Figure 5.6: The resulting obstacle hight limit represented by a color from the color bar for each idler position input as an offset from the default position (left). The obstacle height limit for every actual idler position (right).

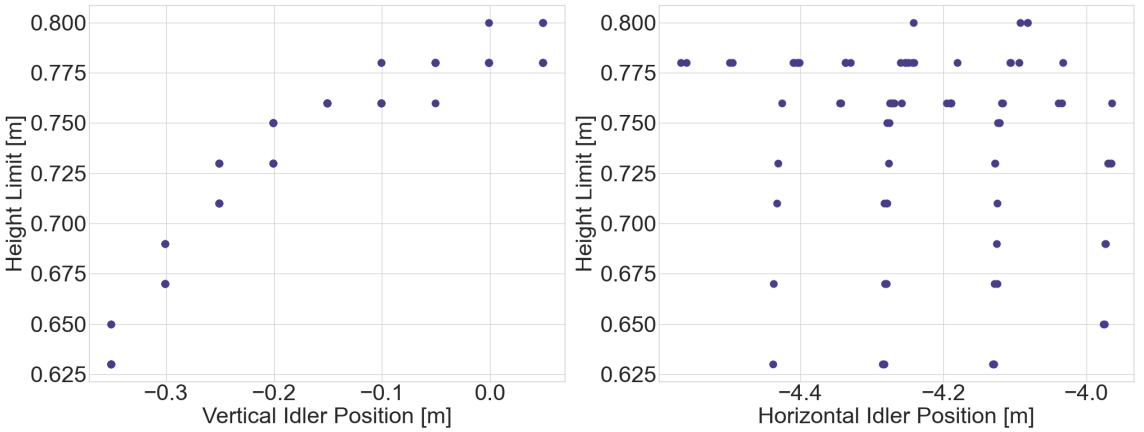

Figure 5.7: The obstacle height limit as a function of the vertical (left) and horizontal (right) idler position.

The most notable pattern is the correlation between the vertical idler position and height limit, shown in the left plot of figure 5.7. An increased idler position results in an increased height limit, this is especially notable for the lower positions. This effect could be caused by a lower idler hitting the ground earlier, when the idler is in contact with the ground a larger lever and therefore momentum is exerted on the vehicle in a way that counteracts the motion of the vehicle.

Regarding the variation of the horizontal position, judging by the right plot of figure 5.7 no clear correlation between the horizontal idler position and climb height limit exists. This is good news for the result from event 1, as the variations in idler position displayed in the right plot of figure 5.5 probably did not affect the resulting height limit in a significant way.

Event 3: The idler position input was varied as in event 2 and the resulting limiting speed

for half-round obstacles was found, as described in section 4.2.1. The speed was increased with 0.5 m/s intervals until a 2.5 g acceleration was measured at the driver's position. The limiting speed was found for half-round obstacles of heights 12, 10 and 8 inches.

The left image of Figures 5.8, 5.9 and 5.10 shows the resulting limiting speed for each idler position input as an offset from the default position. The right image in the same figure shows the limiting speed as a function of the actual idler position, indicated by the black circles. The default position resulted in similar speeds over the 10 and 12 inch obstacles to those found during validation, the difference can be explained by the use of larger steps between the examined speeds during this event.

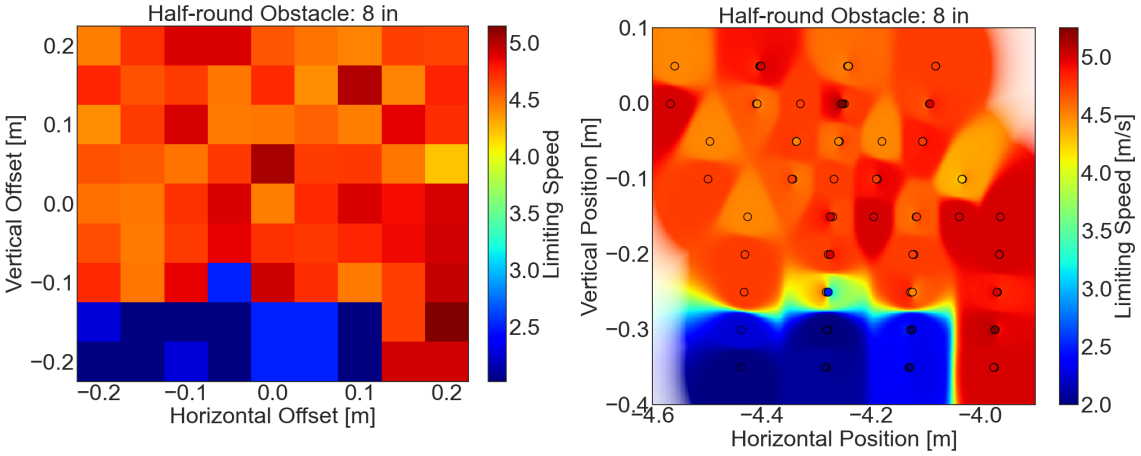

Figure 5.8: The limiting speed over an 8 in half round obstacle for each idler position input as an offset from the default position (left). The limiting speed for the actual idler positions (right).

For the 8 in obstacle the simulation differs a lot from the one performed during the validation, as the evaluation model implements a very high stiffness on the vertical dampers when they exceed 1.27 m/s. This makes a difference on the 8 in obstacle – in contrast to the 10 and 12 in obstacle – as the vehicle speed is high enough for the damper to enter its jounce/stiff region. The resulting limiting speeds shown in figure 5.8 are lower than those found in the validation, it is however closer to the result of vendors C and D, indicating that they might have implemented a similar vertical damper stiffness. From figure 5.8 we can also tell that when the idler is placed far backwards and downwards, the idler hits the obstacle in a similar way to the 10 and 12 in obstacle, leading to a even lower limiting speed.

The limiting speeds over the 10 in obstacle displayed in figure 5.9 show a similar pattern. When the idler is moved up and backwards from its default position, the limiting speed increases as the idler now clears the obstacle.

The limiting speeds over the 12 in obstacle displayed in figure 5.10 show a higher limiting speed when the idler is placed at its highest position, though not as high as in the top left corner. This indicates that the idler partially clears the obstacle when it is placed high up

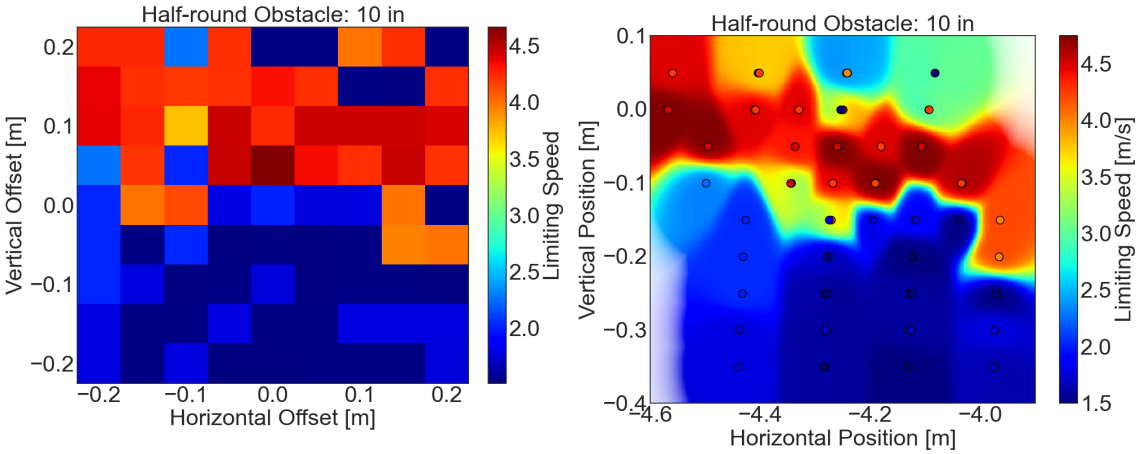

Figure 5.9: The limiting speed over a 10 in half round obstacle for each idler position input as an offset from the default position (left). The limiting speed for the actual idler positions (right).

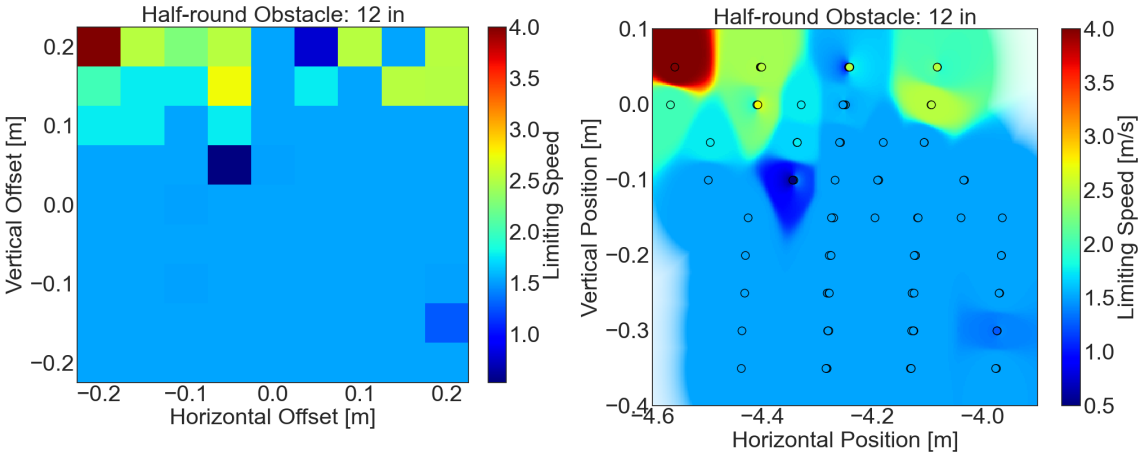

Figure 5.10: The limiting speed over a 12 in half round obstacle for each idler position input as an offset from the default position (left). The limiting speed for the actual idler positions (right).

and completely clears it when also moved far backwards.

Overall the results show that placing the idler higher gives it a bigger chance to not hit the obstacle. When the idler clears the obstacle the resulting limiting speed for all obstacles seems to be around 4.5 m/s using these suspension settings.

5.4 Discussion of Evaluation Results

A natural question regarding the results from the evaluation is: How reliable are they? Can they be used to predict the behavior of an actual vehicle? The answers to those questions depend on a couple of aspects. The first is the validity of the validated model. As already mentioned in section 4.3, the variation of the results presented in the NG-NRMM

report and the lack of empirical data means that no quantitative predictions can be made using the validation model. However, since the model produces results comparable to the best performing models in the NG-NRMM report, this suggests that the general behavior of the model can be trusted when performing tasks similar to those in the validation.

The second requirement for trusting the evaluation results is that the model can be reliably simulated using an unvalidated configuration of the track-assembly. Adequately answering this question would require a more extensive validation study, preferably with a partially parameterized real vehicle. Based on the relatively small validation study presented in this thesis we cannot know the answer for sure. Pointing to our favor however, is the fact that Chrono is an unempirical software shown to be able to produce physically consistent simulations.

Another way to judge the evaluation results would be to compare it to already published work. Unfortunately there is not a lot to be found, especially nothing that allows for a direct comparison. The only notable work found where the author has explicitly varied the sprocket or idler to examine its effect on mobility is Anders Bodin's doctoral thesis "Improving the Mobility Performance of Tracked Vehicles in Deep Snow" [36], where he through both practical experiments and virtual simulations find that lowering the rear idler results in higher drawbar pull and vehicle thrust in deep snow.

If one only considers the evaluation results above, one would with good reason draw the conclusion that placing both the idler and sprocket at a relatively high position is highly beneficial for improving the mobility of a tracked vehicle on rigid terrain. This is of course not always the case, for example over a gap obstacle the opposite is likely true. When also considering deformable terrain, such as snow, we have already seen a counter example in [36].

6

Conclusion

Let us conclude the thesis by stating what was achieved and how it relates to the overarching purpose. First off we can say with certainty that a solution to the problem of static vehicle models in Chrono has been provided with a manager class that allows the user to change model parameters during runtime. Furthermore, with the use of this structure an M113 APC was modeled using publically available data.

The subsequent investigation of simulation parameters and validation of the model, while not completely eliminating all doubts about the validity of the model and simulation, provided insight into the intricacy of vehicle simulations. Specifically, the fact that the same simulation parameters can produce very different behavior for similar vehicle models. Moreover, the remaining uncertainties after the validation highlighted the importance of access to quality empirical data or ideally access to the vehicle one sets out to model. Regardless of any dubiety, the validation results suggested that the vehicle was reasonably modeled and that it exhibits credible behavior under certain circumstances. These being while traversing rigid obstacles at slow speed.

Lastly, the evaluation of different track-assembly geometries showed that the idler and sprocket position has a significant impact on the mobility of the model over rigid obstacles. In particular, that a higher positioned sprocket results in a higher climbable step obstacle and that a higher positioned idler results in both a higher climbable step obstacle as well as a higher limiting speed over 8, 10 and 12 in half round obstacles. How this translates to a general real tracked vehicle requires a more thorough investigation and understanding of the underlying simulation and perhaps more importantly, an indepth validation of the vehicle model including all subsystems.

Considering the aim of investigating how the geometry of the track-assembly of a vehicle affects its mobility, the thesis has provided suggestions of ways that the sprocket and idler position might contribute. With regards to the purpose of aiding in FOI's goal of developing parameterized virtual models of tracked vehicles, the thesis has provided a code structure that overcomes the problem of static vehicle models in the modeling tool Chrono, along with potentially useful insights to the peculiarities of vehicle simulations in general and Chrono in particular.

6.1 Future Work

Given more time there are a handful of actions that would yield immediate improvements upon the methodology and results. These include the simulation of deformable terrain which allows extending the validation to incorporate more of the data available in the NG-NRMM report. Additionally one could expose more of the improvements defining the track-assembly geometry for modification and investigate how they affect vehicle mobility, what readily comes to mind are: torsion arm angles, road wheel radius and initial track tension. Lastly, a big improvement would be further investigation into how the many simulation parameters affect vehicle behavior, including finding appropriate usage of the promising NSC contact formulation.

However, the most obvious step towards further development of the methods and results in this thesis include the introduction of an actual vehicle. The modeling and validation of even a relatively simple vehicle could help sort out some of the uncertainties regarding for example friction and damper behavior. Ideally one would have access to a vehicle similar to that of Anders Bodin in [36] where the parameters of interest can be varied on the actual vehicle. With such a vehicle and corresponding validated model, one could for example make use of various optimization algorithms to optimize the track-assembly for certain tasks and implement this in the actual vehicle.

Bibliography

[1] M. Ogorkiewicz R, *Technology of Tanks I*, 1st ed. Jane's information group Inc., 1991, ISBN: 0-7106-0595-1.

[2] . Project Chrono, *Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems*, Accessed: 2023-09-27. [Online]. Available: http://projectchrono.org.

[3] O. Balling, M. Bradbury, J. Bruce, *et al.*, *AVT-248 Next-Generation NATO Reference Mobility Model (NG-NRMM) Development*. NATO Science and Technology Organisation (STO), 2019, Editors: Jean Dasch and Paramsothy Jayakumar.

[4] A. Tasora, R. Serban, H. Mazhar, *et al.*, "Chrono: An open source multi-physics dynamics engine," in T. Kozubek, Ed., Springer, 2016, pp. 19–49.

[5] *Open-source tools will benefit military and Wisconsin vehicle makers*, Accessed: 2023-09-27. [Online]. Available: https://news.wisc.edu/open-source-tools-will-benefit-military-and-wisconsin-vehicle-makers/.

[6] R. S. Michael Taylor, "Validation of Basic Modeling Elements in Chrono. Technical Report TR-2015-05," in Simulation-Based Engineering Lab University of Wisconsin-Madison, 2015. [Online]. Available: https://sbel.wiscweb.wisc.edu/wp-content/uploads/sites/569/2018/05/TR-2015-05.pdf.

[7] R. S. Michael Taylor and D. Negrut, "Basic Comparison of Chrono::Vehicle and ADAMS/Car. Technical Report TR-2016-15," in Simulation-Based Engineering Lab University of Wisconsin-Madison, 2016. [Online]. Available: https://sbel.wiscweb.wisc.edu/wp-content/uploads/sites/569/2018/05/TR-2016-15.pdf.

[8] R. S. P. J. Arman Pazouki Michał Kwarta Kyle Williams William Likos and D. Negru, "Compliant contact versus rigid contact: A comparison in the context of granular dynamics," *PHYSICAL REVIEW E*, vol. 96, Oct. 2017. DOI: https://doi.org/10.1103/PhysRevE.96.042905.

[9] . Project Chrono, *Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems*, Accessed: 2023-09-27. [Online]. Available: https://projectchrono.org/validation/.

[10] E. J. Haug, *COMPUTER-AIDED KINEMATICS AND DYNAMICS OF MECHANICAL SYSTEMS Volume I: Basic Methods*. Allyn and Bacon, 1989, ISBN: ISBN 0-205-11669-8.

[11] A. Tasora, *Time integration in chrono::engine*, Accessed: 2024-01-22, 2017. [Online]. Available: https://www.projectchrono.org/assets/white_papers/ChronoCore/integrator.pdf.

[12] T. Heyn, M. Anitescu, A. Tasora, D. Negrut, *Using krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation*, Accessed: 2024-04-09. [Online]. Available: https://www.projectchrono.org/tasora/pubblicazioni/IJNME_2012_preprint.pdf.

[13] . Project Chrono, *chrono::ChTimestepperEulerImplicit Class Reference*, Accessed: 2024-01-29. [Online]. Available: https://api.projectchrono.org/8.0.0/classchrono_1_1_ch_timestepper_euler_implicit.html.

[14] . Project Chrono, *chrono::ChTimestepperEulerImplicitLinearized Class Reference*, Accessed: 2023-11-8. [Online]. Available: https://api.projectchrono.org/8.0.0/classchrono_1_1_ch_timestepper_euler_implicit_linearized.html.

[15] . Project Chrono, *chrono::ChTimestepperEulerImplicitProjected Class Reference*, Accessed: 2023-11-8. [Online]. Available: https://api.projectchrono.org/8.0.0/classchrono_1_1_ch_timestepper_euler_implicit_projected.html.

[16] M. Benzi, G. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, May 2005. DOI: 10.1017/S0962492904000212.

[17] J. Fleischmann, *Dem-pm contact model with multi-step tangential contact displacement history*, Accessed: 2024-01-29, 2015. [Online]. Available: https://sbel.wiscweb.wisc.edu/wp-content/uploads/sites/569/2018/05/TR-2015-06.pdf.

[18] R. S. D. Negrut, *Posing multibody dynamics with friction and contact as a differential complementarity problem*, Accessed: 2024-01-30, 2016. [Online]. Available: https://sbel.wiscweb.wisc.edu/wp-content/uploads/sites/569/2018/05/TR-2016-12.pdf.

[19] *Overview of vehicle modeling and simulation*, Accessed: 2024-01-11. [Online]. Available: https://api.projectchrono.org/vehicle_overview.html.

[20] *Tracked vehicles*, Accessed: 2024-01-11. [Online]. Available: https://api.projectchrono.org/tracked_vehicle.html.

[21] *Creating easy iteration vehicle model*, Accessed: 2024-01-12. [Online]. Available: https://groups.google.com/g/projectchrono/c/YOVnPTJzeiE/m/eBpR2vhNAAAJ.

[22] S. R. T. M. M. Daniel and N. Dan, *Ng-nrmm phase i benchmarking: Chrono tracked vehicle simulation results summary*, Aug. 2016.

[23] P. A. Dudziński and J. Chołodowski, "A method for predicting the internal motion resistance of rubber-tracked undercarriages, pt. 1. a review of the state-of-the-art methods for modeling the internal resistance of tracked vehicles," *Journal of Terramechanics*, vol. 96, pp. 81–100, 2021, ISSN: 0022-4898. DOI: https://doi.org/10.1016/j.jterra.2021.02.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022489821000173.

[24] *Chrono::chsolverpsor class reference*, Accessed: 2023-11-8. [Online]. Available: https://api.projectchrono.org/8.0.0/classchrono_1_1_ch_solver_p_s_o_r.html.

[25] *Chrono::chsolverbb class reference*, Accessed: 2023-11-8. [Online]. Available: https://api.projectchrono.org/8.0.0/classchrono_1_1_ch_solver_b_b.html.

[26] R. G. R. Serban, *Demonstration program for marder vehicle on rigid terrain.* Accessed: 2024-02-21, 2022. [Online]. Available: https://github.com/projectchrono/chrono/blob/release/8.0/src/demos/vehicle/tracked_models/demo_VEH_Marder.cpp.

[27] R. Serban, *Demonstration program for m113 vehicle on scm terrain*, Accessed: 2024-02-21, 2022. [Online]. Available: https://github.com/projectchrono/chrono/blob/release/8.0/src/demos/vehicle/tracked_models/demo_VEH_M113_DefSoil.cpp.

[28] *Young's modulus, tensile strength and yield strength values for some materials*, Accessed: 2024-02-23. [Online]. Available: https://www.engineeringtoolbox.com/young-modulus-d_417.html.

[29] *Test Operations Procedure (TOP)1-1-014 Ride Dynamics*, 2005.

[30] *Test Operations Procedure (TOP)2-2-611A Standard Obstacles*, 2022.

[31] *Wood, panel and structural timber products - mechanical properties*, Accessed: 2024-01-09. [Online]. Available: https://www.engineeringtoolbox.com/timber-mechanical-properties-d_1789.html.

[32] R. Hanxue, *Tank terrain trials: T-72, mt-lb, centurion, pbv 302*, Accessed: 2024-02-22, 2015. [Online]. Available: https://www.youtube.com/watch?v=BjkWgek6UXU&t=768s.

[33] I. BAE Systems, *Cv90: Superior mobility in the toughest terrains*, Accessed: 2024-02-22, 2021. [Online]. Available: https://www.youtube.com/watch?v=xQDq47i2sjE.

[34] S. Crater, *Danish army's leopard 2a7 during obstacle course live fire exercises*, Accessed: 2024-02-22, 2021. [Online]. Available: https://youtu.be/x1pVRnvuC14?feature=shared&t=39.

[35] P. Wang, X. Rui, and H. Yu, "Study on dynamic track tension control for high-speed tracked vehicles," *Mechanical Systems and Signal Processing*, vol. 132, pp. 277–292, 2019, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2019.06.031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327019304224.

[36] A. Bodin, "Improving the mobility performance of tracked vehicles in deep snow," 2002.

[37] A. Tasora, *Rotations in chrono::engine*, Accessed: 2024-01-22, 2020. [Online]. Available: https://www.projectchrono.org/assets/white_papers/ChronoCore/rotations.pdf.

[38] . Project Chrono, *Chrono 3.0.0 training material*, Accessed: 2024-05-05. [Online]. Available: https://api.projectchrono.org/development/tutorial_slides_300.html.

[39] A. Kissel, J. Taves, and D. Negrut, "Constrained Multibody Kinematics and Dynamics in Absolute Coordinates: A Discussion of Three Approaches to Representing Rigid Body Rotation," *Journal of Computational and Nonlinear Dynamics*, vol. 17, no. 10, p. 101 008, Aug. 2022, ISSN: 1555-1415. DOI: 10.1115/1.4055140. [Online]. Available: https://doi.org/10.1115/1.4055140.

A

Quaternions and Rotations

This chapter introduces quaternions and how they can be used to encode rotations and orientation, to the extent needed for the theory presented in this thisis. For a more indepth walkthrough of the theory see [10] and [37].

The quaternion number system is an four dimensional extension of the complex number systems with three imaginary axises. The algebra of quatenions will be denoted $\mathbb{H}$. A quaternion $\boldsymbol{q} \in \mathbb{H}$ can be expressed as

$$\boldsymbol{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$$

where $a, b, c, d \in \mathbb{R}$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are unit vectors along the imaginary axises with the properties:

$$\begin{aligned}
\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 &= -1 \\
\boldsymbol{ij} &= -\boldsymbol{ji} = \boldsymbol{k} \\
\boldsymbol{jk} &= -\boldsymbol{kj} = \boldsymbol{i} \\
\boldsymbol{ki} &= -\boldsymbol{ik} = \boldsymbol{j} \\
\boldsymbol{ijk} &= -1.
\end{aligned}$$

Another useful way to represent a quaternion is by spliting up the scalar part $s \in \mathbb{R}$ and imaginary vector part $\boldsymbol{v} \in \text{Im}\mathbb{H}$ as

$$\boldsymbol{q} = [s, \boldsymbol{v}]. \tag{A.1}$$

This representation allows us to express addition and multiplication of two quaternions $\boldsymbol{a}, \boldsymbol{b}$ as

$$\begin{aligned}
\boldsymbol{a} \pm \boldsymbol{b} &= [s_a \pm s_b, \boldsymbol{v}_a \pm \boldsymbol{v}_b] \\
\boldsymbol{ab} &= [s_a s_b - \boldsymbol{v}_a \cdot \boldsymbol{v}_b, s_a\boldsymbol{v}_b + s_b\boldsymbol{v}_a + \boldsymbol{v}_a \times \boldsymbol{v}_b]
\end{aligned}$$

where $(\cdot)$ and $(\times)$ are the dot product and cross product of $\mathbb{R}^3$.

Furthermore we have the properties

$$\begin{aligned}
\boldsymbol{q}^* &= q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k} \\
|\boldsymbol{q}| &= \sqrt{\boldsymbol{q}\boldsymbol{q}^*} \\
|\boldsymbol{q}| &\geq 0 \\
|\boldsymbol{q_1}\boldsymbol{q_2}| &= |\boldsymbol{q_1}||\boldsymbol{q_2}|
\end{aligned} \tag{A.2}$$

Let us continue by considering Euler's theorem about 3D rotations.

Euler's Theorem as stated in [10]: If the origins of two right-hand Cartesian reference frames coincide, then they may be brought into coincidence by a single rotation about some axis.

The implications of the theorem is that a rotation in 3D space can be specified by four scalars, where three are used to specify the axis of rotation $\boldsymbol{u}$ and the fourth to specify the amount to rotate ϕ. One way to encode this is by using a quaternion, which has the advantage of being more compact and numerically stable than the usual rotation matrix.

This is achieved using the transformation $\mathbb{H} \ni \boldsymbol{\rho} \rightarrow \boldsymbol{\rho'} \in \mathbb{H}$:

$$\boldsymbol{\rho'} = \boldsymbol{q}\boldsymbol{\rho}\boldsymbol{q}^*. \tag{A.3}$$

From the property (A.2) we have

$$|\boldsymbol{\rho'}| = |\boldsymbol{q}\boldsymbol{\rho}\boldsymbol{q}^*| = |\boldsymbol{q}||\boldsymbol{\rho}||\boldsymbol{q}^*|$$

which means that when using a unit length quaternion $|\boldsymbol{q}| = 1$ the length of $\boldsymbol{\rho}$ is unchanged. Thus (A.3) acts as a rotation of the $\boldsymbol{\rho}$ quaternion. In the case where $\boldsymbol{\rho} = [0, \boldsymbol{v}]$ is purly imaginary the transformed $\boldsymbol{\rho'} = [0, \boldsymbol{v'}]$ is also purly imaginary, meaning that the transform (A.3) can be used to express a rotation of a 3D vector $\boldsymbol{v}$ to $\boldsymbol{v'}$. The corresponding matrix A describing the same rotation $\boldsymbol{v'} = A\boldsymbol{v'}$ would be:

$$A = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(-q_1q_0 + q_2q_3) \\ 2(q_1q_3 - q_2q_0) & 2(q_1q_0 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}.$$

To achieved a specific rotation of ϕ degrees around the unit axis $\boldsymbol{u}$ we use a quaternion $\boldsymbol{q}$ with the values

$$\begin{aligned} q_0 &= \cos\frac{\phi}{2} \\ q_1 &= u_x \sin\frac{\phi}{2} \\ q_1 &= u_y \sin\frac{\phi}{2} \\ q_1 &= u_z \sin\frac{\phi}{2}. \end{aligned}$$

Such a quaternion can also be used to describe the orientation of a rigid body as long as a reference axis is specified.

The quaternion exponential function $\exp(\boldsymbol{q})$ used in (2.9) to express an incremental rotational update expressed in angular velocity is defined using the notation of (A.1) as

$$\exp(\boldsymbol{q}) = \exp\big([s, \boldsymbol{v}]\big) = e^s \left[\cos|\boldsymbol{v}|, \frac{\boldsymbol{v}}{|\boldsymbol{v}|}\sin|\boldsymbol{v}|\right]$$

B

Algebraic Constaints

This chapter will treat the basics of kinematic constraints and constraint jaccobian of a system and is based on the Project Chrono documentation [38]. A more detailed explaination can be found in [10] and [39]. The chapter will use the notation established in section 2.4 and appendix A.

In the context of multibody simulations, kinematic constraints are represented and enforced by algebraic constraint equations (ACE) that specify the allowed relative position and orientation of pairs of rigid bodies connected by joints. There are four basic shuch equations that can be combined to make most other useful joints, they are presented in table B.1.

Table B.1: The four basic geometric constraints

Name	**Description**
DP1	Specifies the dot product between two vectors on two different bodies
DP2	Specifies the dot product between a vector on a body and a vector between two bodies
D	Specifies the distance between two bodies
CD	Specifies the difference in coordinates between two points on two bodies

To describe the reaction forces from these constraints the partial derivatives $\mathbf{\Phi_q}$ are needed. When using quaternions to express the orientation of the bodies this is straightforward. However when using orientation matrices one instead need to use the coefficient matrix $\bar{\mathbf{\Pi}}(\mathbf{\Phi})$ that multiplies $\bar{\boldsymbol{\omega}}$ in the expression of the time derivative $\dot{\mathbf{\Phi}}$.

DP1
The ACE for the DP1 constraint is

$$\Phi^{DP1}(\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{a}}_j, f(t)) = \bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{A}_j \bar{\boldsymbol{a}}_j - f(t) = 0 \tag{B.1}$$

here the ACE is expressed in the local reference frames using orientation matrices. When using quaternions however, it is advantageous to express the ACE in the global reference which is simply

$$\Phi^{DP1}(\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{a}}_j, f(t)) = \boldsymbol{a}_i^T \boldsymbol{a}_j - f(t) = 0 \tag{B.2}$$

Figure B.1 shows the two bodies and the vectors $\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{a}}_j$ in the local reference frame (L-RF).

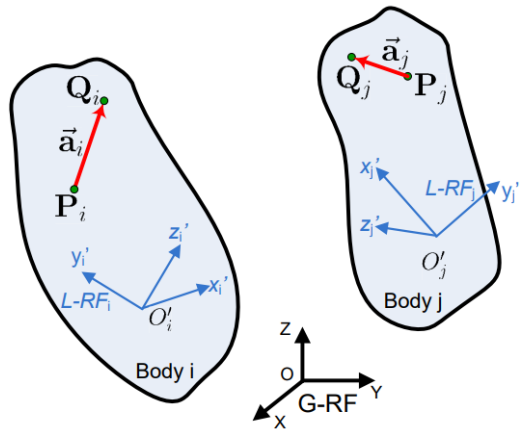

Figure B.1: A graphical representation of two bodies i and j constrained by a DP1 joint. The dot product between the vector $\bar{\boldsymbol{a}}_i$ from point $\boldsymbol{P}_i$ to $\boldsymbol{Q}_i$ and the vector $\bar{\boldsymbol{a}}_j$ from point $\boldsymbol{P}_j$ to $\boldsymbol{Q}_j$ is enforced to equal a value $f(t)$. Figure from [2].

To calculate the partial derivatives $\boldsymbol{\Phi}_{\boldsymbol{r}}$ and $\boldsymbol{\Phi}_{\boldsymbol{\rho}}$ of the constraint we look to [10] and find the matrix $\boldsymbol{B}(\boldsymbol{\rho}, \bar{\boldsymbol{a}}) \in \mathbb{R}^{3\times4}$ as a function of a positional vector $\bar{\boldsymbol{a}} \in \mathbb{R}^3$ expressed in a body local frame and a rotational quaternion $\boldsymbol{\rho} \in \mathbb{R}^4$ represented by a vector of four real numbers. With the quaternion representation $\boldsymbol{\rho} = [s, \boldsymbol{v}]$ the matrix is defined as:

$$\boldsymbol{B}(\boldsymbol{\rho}, \bar{\boldsymbol{a}}) = 2\big[(s\boldsymbol{I}_3 + \tilde{\boldsymbol{v}})\bar{\boldsymbol{a}}, \boldsymbol{v}\bar{\boldsymbol{a}}^T - (s\boldsymbol{I}_3 + \tilde{\boldsymbol{v}})\tilde{\bar{\boldsymbol{a}}}\big].$$

As is shown in [10] the partial derivative of a body fixed positional vector $\boldsymbol{a}$ – expressed in a global refrence frame – with respect to the orientational quaternion of the body is

$$\boldsymbol{a}_p = \boldsymbol{B}(\boldsymbol{\rho}, \bar{\boldsymbol{a}}).$$

This result allows us to calculate the partial derivatives of the ACE (B.2) as

$$\frac{\partial \Phi^{DP1}}{\partial \boldsymbol{\rho}_i} = \boldsymbol{a}_j^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{a}}_i)$$
$$\frac{\partial \Phi^{DP1}}{\partial \boldsymbol{\rho}_j} = \boldsymbol{a}_i^T \boldsymbol{B}(\boldsymbol{\rho}_j, \bar{\boldsymbol{a}}_j)$$

and since (B.2) is independant of the origins $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ of the bodies we have

$$\frac{\partial \Phi^{DP1}}{\partial \boldsymbol{r}_i} = \boldsymbol{0}_{1\times3}$$
$$\frac{\partial \Phi^{DP1}}{\partial \boldsymbol{r}_j} = \boldsymbol{0}_{1\times3}$$

To get the coefficient matrix $\bar{\mathbf{\Pi}}(\mathbf{\Phi})$ we look at the time derivative of the ACE expressed in the local reference frames (B.1)

$$\dot{\Phi}^{DP1}(\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{a}}_j, f(t)) = \bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \dot{\boldsymbol{A}}_j \bar{\boldsymbol{a}}_j + \bar{\boldsymbol{a}}_j^T \boldsymbol{A}_j^T \dot{\boldsymbol{A}}_i \bar{\boldsymbol{a}}_i - \dot{f}(t) \tag{B.3}$$

looking at [10] we find the formulas

$$\dot{\boldsymbol{A}} = \boldsymbol{A}\tilde{\bar{\omega}}$$

and

$$\tilde{\boldsymbol{a}}\boldsymbol{b} = -\boldsymbol{b}\tilde{\boldsymbol{a}}$$

which allows us to rewrite (B.3) as

$$\dot{\Phi}^{DP1}(\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{a}}_j, f(t)) = -\bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{A}_j \tilde{\bar{\boldsymbol{a}}}_j \bar{\omega}_j - \bar{\boldsymbol{a}}_j^T \boldsymbol{A}_j^T \boldsymbol{A}_i \tilde{\bar{\boldsymbol{a}}}_i \bar{\omega} - \dot{f}(t)$$

where the coefficient matrices are

$$\bar{\mathbf{\Pi}}_i(\mathbf{\Phi}^{DP1}) = -\bar{\boldsymbol{a}}_j^T \boldsymbol{A}_j^T \boldsymbol{A}_i \tilde{\bar{\boldsymbol{a}}}_i$$
$$\bar{\mathbf{\Pi}}_j(\mathbf{\Phi}^{DP1}) = -\bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{A}_j \tilde{\bar{\boldsymbol{a}}}_j$$

DP2

Next up is the DP2 constraint wich introduces a vector $\boldsymbol{d}_{ij}$ between point $\boldsymbol{P}_i$ on body i and point $\boldsymbol{Q}_j$ on body j. This vector can be expressed as

$$\boldsymbol{d}_{ij} = \boldsymbol{r}_j + \boldsymbol{A}_j \bar{\boldsymbol{s}}_j^Q - \boldsymbol{r}_i - \boldsymbol{A}_i \bar{\boldsymbol{s}}_i^P = \boldsymbol{r}_j + \boldsymbol{s}_j^Q - \boldsymbol{r}_i - \boldsymbol{s}_i^P$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_i$ are the origins of the local reference frames for body i and j. The vector $\bar{\boldsymbol{s}}_i^P$ is the vector from the local reference frame origin of body i to the point $\boldsymbol{P}_i$ on body i and $\bar{\boldsymbol{s}}_j^Q$ is the vector from the local reference frame origin of body j to the point $\boldsymbol{Q}_j$ on body j. Figure B.2 shows the bodies and the relevant vectors for the DP2 constraint.

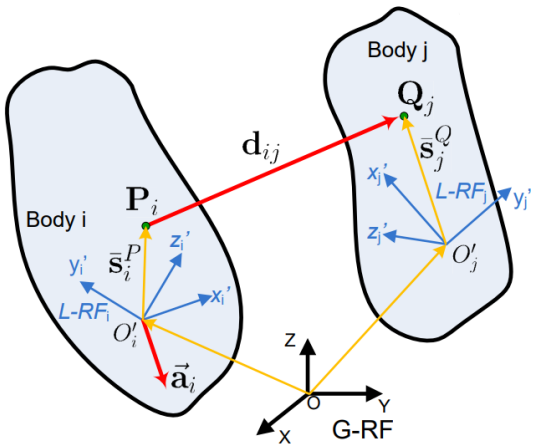

Figure B.2: A graphical representation of two bodies i and j constrained by a DP2 joint. The dot product between the vector $\bar{\boldsymbol{a}}_i$ and the vector $\boldsymbol{d}_{ij}$ between point $\boldsymbol{P}_i$ on body i and point $\boldsymbol{Q}_j$ on body j is enforced to equal a value $f(t)$. Figure from [2].

The ACE of DP2 is

$$\Phi^{DP2}(\bar{\boldsymbol{a}}_i, \boldsymbol{d}_{ij}, f(t)) = \bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{d}_{ij} - f(t) = \boldsymbol{a}_i^T \boldsymbol{d}_{ij} - f(t) = 0$$

the partial derivatives are

$$\frac{\partial \Phi^{DP2}}{\partial \boldsymbol{r}_i} = -\boldsymbol{a}_i^T \qquad \frac{\partial \Phi^{DP2}}{\partial \boldsymbol{\rho}_i} = \boldsymbol{d}_{ij}^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{a}}_i) - \boldsymbol{a}_i^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{s}}_i^P)$$
$$\frac{\partial \Phi^{DP2}}{\partial \boldsymbol{r}_j} = \boldsymbol{a}_i^T \qquad \frac{\partial \Phi^{DP2}}{\partial \boldsymbol{\rho}_i} = \boldsymbol{d}_{ij}^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{a}}_i) - \boldsymbol{a}_i^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{s}}_i^P)$$

and the coefficient matrices are.

$$\bar{\boldsymbol{\Pi}}_i(\boldsymbol{\Phi}^{DP2}) = \bar{\boldsymbol{a}}_i^T \tilde{\bar{\boldsymbol{s}}}_i^P - \boldsymbol{d}_{ij}^T \boldsymbol{A} \tilde{\bar{\boldsymbol{a}}}_i$$
$$\bar{\boldsymbol{\Pi}}_j(\boldsymbol{\Phi}^{DP2}) = -\bar{\boldsymbol{a}}_i^T \boldsymbol{A}_i^T \boldsymbol{A}_j \tilde{\bar{\boldsymbol{s}}}_j^Q$$

D

The third basic constraint is the distance constraint that specify the distance between two points on two different bodies. The constraint is visualized in figure B.3.

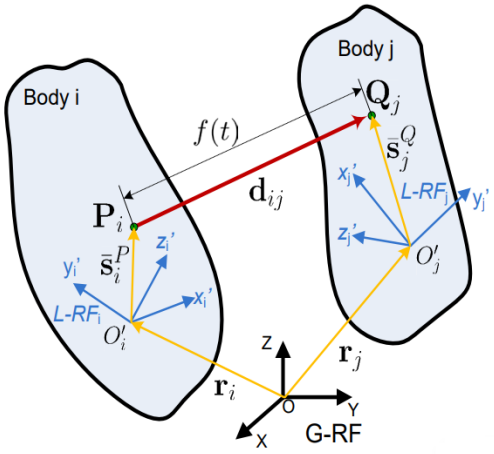

Figure B.3: A graphical representation of two bodies i and j constrained by a D joint. The distance between the point $\boldsymbol{P}_i$ on body i and the pont $\boldsymbol{Q}_j$ on body j is enforced to equal a value $f(t)$. Figure from [2].

The ACE of D is

$$\Phi^D(\bar{\boldsymbol{s}}_i^P, \bar{\boldsymbol{s}}_j^Q, f(t)) = (\boldsymbol{r}_j + \boldsymbol{A}_j \bar{\boldsymbol{s}}_j^Q - \boldsymbol{r}_i - \boldsymbol{A}_i \bar{\boldsymbol{s}}_i^P)^T (\boldsymbol{r}_j + \boldsymbol{A}_j \bar{\boldsymbol{s}}_j^Q - \boldsymbol{r}_i - \boldsymbol{A}_i \bar{\boldsymbol{s}}_i^P) - f^2(t) = \boldsymbol{d}_{ij}^T \boldsymbol{d}_{ij} - f^2(t)$$

the partial derivatives are

$$\frac{\partial \Phi^D}{\partial \boldsymbol{r}_i} = -2\boldsymbol{d}_{ij}^T \qquad \frac{\partial \Phi^D}{\partial \boldsymbol{\rho}_i} = -2\boldsymbol{d}_{ij}^T \boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{s}}_i^P)$$
$$\frac{\partial \Phi^D}{\partial \boldsymbol{r}_j} = 2\boldsymbol{d}_{ij}^T \qquad \frac{\partial \Phi^D}{\partial \boldsymbol{\rho}_j} = 2\boldsymbol{d}_{ij}^T \boldsymbol{B}(\boldsymbol{\rho}_j, \bar{\boldsymbol{s}}_j^Q)$$

and the coefficient matrices are.

$$\bar{\mathbf{\Pi}}_i(\mathbf{\Phi}^D) = 2\boldsymbol{d}_{ij}^T \boldsymbol{A}_i \tilde{\bar{\boldsymbol{s}}}_i^P$$
$$\bar{\mathbf{\Pi}}_j(\mathbf{\Phi}^D) = -2\boldsymbol{d}_{ij}^T \boldsymbol{A}_j \tilde{\bar{\boldsymbol{s}}}_j^Q$$

CD

The last basic constraint is the coordinate difference constraint with the ACE

$$\Phi^{CD}(\boldsymbol{c}, \bar{\boldsymbol{s}}_i^P, \bar{\boldsymbol{s}}_j^Q, f(t)) = \boldsymbol{c}^T(\boldsymbol{r}_j + \boldsymbol{A}_j \bar{\boldsymbol{s}}_j^Q - \boldsymbol{r}_i - \boldsymbol{A}_i \bar{\boldsymbol{s}}_i^P) - f^2(t) = \boldsymbol{c}^T \boldsymbol{d}_{ij} - f^2(t).$$

the global vector $\boldsymbol{c}$ is choosen as one of the main axises of the global referance frame thus the constraint effectively specifies the difference between the x,y or z value between two points on two bodies, hence the name of the constraint. The partial derivatives are

$$\frac{\partial \Phi^{CD}}{\partial \boldsymbol{r}_i} = -\boldsymbol{c}^T \qquad \frac{\partial \Phi^{CD}}{\partial \boldsymbol{\rho}_i} = -\boldsymbol{B}(\boldsymbol{\rho}_i, \bar{\boldsymbol{s}}_i^P)$$
$$\frac{\partial \Phi^{CD}}{\partial \boldsymbol{r}_j} = \boldsymbol{c}^T \qquad \frac{\partial \Phi^{CD}}{\partial \boldsymbol{\rho}_j} = \boldsymbol{B}(\boldsymbol{\rho}_j, \bar{\boldsymbol{s}}_j^Q)$$

and the coefficient matrices are.

$$\bar{\mathbf{\Pi}}_i(\mathbf{\Phi}^{CD}) = \boldsymbol{c}^T \boldsymbol{A}_i \tilde{\bar{\boldsymbol{s}}}_i^P$$
$$\bar{\mathbf{\Pi}}_j(\mathbf{\Phi}^{CD}) = -\boldsymbol{c}^T \boldsymbol{A}_j \tilde{\bar{\boldsymbol{s}}}_j^Q$$

By combining these four basic constraints a large number of more complicated joints can be defined and they can all be broken down to these basic cases when simulating the system.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY