Design, Prototyping, and Testing of an In-Wheel Suspension System

by

Mohsen Azimi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2009

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis presents a study of a novel suspension system which is placed inside a vehicle's wheel. The In-wheel suspension system isolates the sprung mass from excitations similar to conventional suspension systems. In traditional suspension systems the isolation is provided by spacious and complicated mechanisms, and mainly in the vertical direction. However, the in-wheel suspension system, not only fits the suspension mechanism inside the unused space between a wheel’s rim and hub, but also allows for isolation both in vertical and horizontal directions.

The main focus of this thesis is to study, investigate, and show the feasibility of applying such suspension system to a vehicle. This research is conducted on low speed, low load, and non-powered vehicles such as hand trucks and baby strollers. This helps to escape from the complications of a complex system like a road vehicle. It also demonstrates the versatility of the in-wheel suspension idea. The objective of the project is to scrutinize a simple but practical in-wheel suspension system and demonstrate its applicability.

The research begins with the dynamics modeling of an in-wheel suspension system. This suspension has been previously developed at the University of Waterloo for a wheelchair. The dynamics model evaluates the response of the suspension system and investigates the influence of various design parameters on the in-wheel suspension. The study is then continued to improve the design by replacing its rigid mechanism links with optimized compliant structures. This reduces the system's complexity and weight while boosting its performance. Furthermore, a general optimization code is developed to design and optimize flexible members for in-wheel suspension systems. The optimization program is then used to design and optimize two prototypes for hand trucks. Finally, the in-wheel suspension system for a hand truck is tested and evaluated. The experimental results also verify the simulation results and verify the developed optimization design program.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Amir Khajepour, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his constant support and assistance which made this work possible.

I would also like to acknowledge Ontario Centres of Excellence (OCE), Mindmatter Inc. and Cooper-Standard Automotive for their financial and technical support.

A very special thanks goes to Jason Benning, Joe M. Mihalic, Amin Eshraghi, Mohammadreza Noban, and Jafar Al Bin Mousa for their kind assistance.

I must also acknowledge, Alireza Kasaiezadeh, Mehrdad Iravani, Saman Hosseini, Amin Kamalzadeh, and Babak Ebrahimi for providing a warm and friendly atmosphere which made my graduate studies a memorable experience.

I would also like to thank my family for the support they provided me through my entire life without whose love, and encouragement I would not have finished this thesis.

Dedication

This thesis is dedicated to my parents and grandparents.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The invention of the first four-stroke cycle gasoline engine automobile by Karl Benz in 1985 (Stein, 1967) has changed transportation technology forever. Since then, the addition of modern engines, brake systems, comfortable suspension systems, and various electro-mechanical instruments has evolved automobile to a safer, faster, and more comfortable mean of transportation. The current state of the art automobile design seems to be flawless; however high energy prices and environmental issues challenge engineers to further improve the automobile design.

One of the major subsystems in a modern passenger car is the suspension system. The suspension system of a road vehicle refers to the assembly between the chassis and the wheels. It transfers forces and moments from the contact patch to the chassis. Vehicle suspension systems are designed to provide ride comfort, road-holding, handling, and directional performance. Different forms of suspension systems, such as McPherson, double wishbone and multi-link suspensions, have been adapted in passenger vehicles (Reimpell, et al., 2001). Although the current suspension systems effectively isolate the sprung mass from road excitations in vertical direction, but the sprung mass' horizontal connection to the ground is almost rigid. The slight horizontal isolation is achieved by suspension mounts and tire (Reimpell, et al., 2001). Furthermore, traditional suspension systems usually consist of spacious three dimensional mechanisms with heavy components. The higher suspension system mass (part of the un-sprung mass) not only decrease the isolation effectiveness (Dixon, 1996), but also results in lower fuel efficiency. A study of European vehicles weighing less than 1000kg shows that a 10% increase in vehicles' mass, increases the fuel consumption by 7-8% (Why has car-fleet specific fuel consumption not shown any decrease since 1990? Quantitative analysis of Dutch passenger car-fleet specific fuel consumption, 2001).

In order to overcome the limitations of conventional suspension systems, a new concept of the In-wheel Suspension System (IWS) is proposed. The IWS concept not only improves the horizontal isolation but also eliminates the spacious suspension mechanisms by embedding the suspension system inside the vehicle's wheel. The development of IWS is based on the idea of fitting the spring and damping elements inside a vehicle’s wheel. Depending on whether the suspension rotates with the wheel or not, the IWS can be classified to: Rotating and Non-Rotating IWS (Figure 1.1). From another point of view, if the IWS is placed inside a driving wheel, the IWS can be classified to

Powered or Non -Powered IWS. A Powered IWS requires an infinite torsional stiffness to transfer tracking or breaking forces to the wheel.

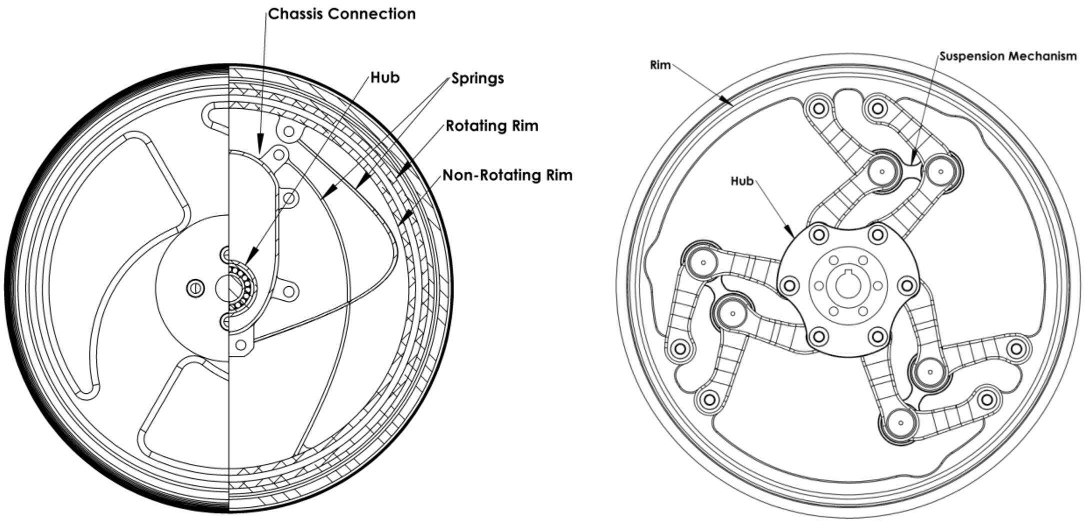


Figure 1.1 - In-wheel suspension system (left: Non-Rotating, right: Rotating)

This thesis focuses on the study and design of rotating non-powered IWSs for low speed vehicles. The rotating in-wheel suspension system, like other suspensions should provide desired stiffness and damping rates. It should also allow the maximum possible wheel travel. For the rotating IWS in addition to the conventional suspension properties, the stiffness fluctuation should be considered as well. The rotation of the suspension system which is embedded inside the vehicle's wheel, changes the orientation of the suspension elements at each rotation angle of the wheel. Consequently, the change in the suspension component's orientation can lead to undesired stiffness fluctuations. These fluctuations are sensed by the sprung mass as periodic vertical or horizontal vibrations when moving on a flat surface. The major challenge in the IWS design is the space limitation. The suspension system should fit inside the vehicle's wheel without altering its outer dimensions. This design requires selection of proper materials for flexible suspension components and optimization of the shape of the components for most energy absorbance efficiency. Furthermore, the design should be simple, easy to manufacture, and also retrofit.

This thesis follows a systematic approach in studying and designing rotating in-wheel suspension systems for low speed, low load vehicles such as hand trucks, wheel chairs, bicycles, and baby strollers. The modeling and optimization work is carried out using well known engineering packages such as MATLAB, ADAMS, and ANSYS.

The proceeding chapters in this thesis are organized as follows:

A literature review on the existing isolating wheel designs is presented in Chapter 2. In general, most of these designs are for eliminating inflatable pneumatic tires.

A multi-body dynamics model of a rotating in-wheel suspension is studied in Chapter 3. This chapter includes derivation of equations of motion for a mechanism based rotating in-wheel suspension. This suspension has been previously designed for a wheelchair using static calculation at the University of Waterloo. The equations of motions are solved and investigated for the influence of various design parameters to better understand rotating suspension systems.

In Chapter 4, the rigid members in the mechanism based in-wheel suspension system discussed in Chapter 3 are replaced with flexible members. This decreases the number of components, the suspension system's weight, and also the design complexity. An optimization tool is also developed in MATLAB that can be used for designing flexible rotating in-wheel suspension systems.

Chapter 5 presents the implementation of the flexible rotating suspensions introduced in Chapter 4. Two prototypes are optimized, manufactured, and tested to verify the simulation results.

Chapter 2
Literature Review

2.1 In-wheel suspension systems

For any road vehicle, a common method to isolate passengers from undesirable excitation caused by road roughness or onboard sources is the use of a suspension system (Gillespie, 1992). Suspension systems have been utilized on different types of vehicles, such as motorcycles, passenger cars, trucks and bikes. Suspension is basically an assembly of spring and damper components used to reduce the transferred excitations to a vehicle's chassis. The spring element can be a coil spring, leaf spring, torsion bar, or an air spring. The damper element is usually a shock absorber. Different forms of suspension systems such as McPherson, double wishbone, and multi-link suspensions have been adapted to passenger vehicles. In addition to the suspension system, the stiffness and damping properties of tires, which are usually pneumatic, help to partially alleviate shocks and vibrations. The traditional construction and configuration of suspension systems with spacious two or three dimensional mechanisms has remained unchanged for a long time.

Although road excitation may be in the vertical direction, longitudinal direction, or the lateral direction (SAE, 1978), current suspension systems mainly isolate vertical forces and disturbances. In many situations, isolation in more than one direction can improve safety and comfort significantly. For instance, the addition of a suspension system that is capable of reacting to longitudinal forces in wheelchairs, bicycles and motorcycles, can reduce the incidents resulting from tipping over upon hitting obstacles. Furthermore, conventional suspension systems usually occupy a significant amount of space. Such space could be used for passenger comfort or the reduction of the overall size and weight of the vehicle. Moreover, the existing suspension systems are difficult to access, making maintenance and repair difficult. Relatively heavy and complex components reduce the vehicle's fuel efficiency (Why has car-fleet specific fuel consumption not shown any decrease since 1990? Quantitative analysis of Dutch passenger car-fleet specific fuel consumption, 2001).

The IWS is a novel concept in suspension system design applicable to wheelchairs, bicycles, motorcycles, automobiles, trucks, and even airplane landing gears. Previous experimental studies at the University of Waterloo on wheelchairs has shown a tremendous improvement in the level of vibration and comfort in wheelchairs and also a good potential for the IWS concept to become a strong alternative to conventional suspension systems in other applications (Khajepour, 2007). In addition to the studies at the University of Waterloo, there have been other efforts to design flexible isolating wheels. Some of these designs are reviewed in this section. Most of these designs focus on replacing the inflatable pneumatic tire with a flexible maintenance free wheel. Michelin's TWEEL is the first non-pneumatic tire announced. Figure 2.1 shows Michelin's TWEEL. TWEEL is a revolutionary non-pneumatic wheel first announced by Michelin in 2005. TWEEL consists of a solid inner hub which mounts to the axle. The hub is enclosed by polyurethane spokes arrayed in a pattern of wedges. The outer edge of the tire, the part that contacts with the road, is formed by stretching a shear band across the spokes. The strength of the spokes combined with the tension of the shear band replaces the air pressure of a traditional tire. Although no specific data is given by Michelin on the mechanical properties and performance of TWEEL; vibration, noise, and heat generation are known as the shortcomings of TWEEL at higher speeds.

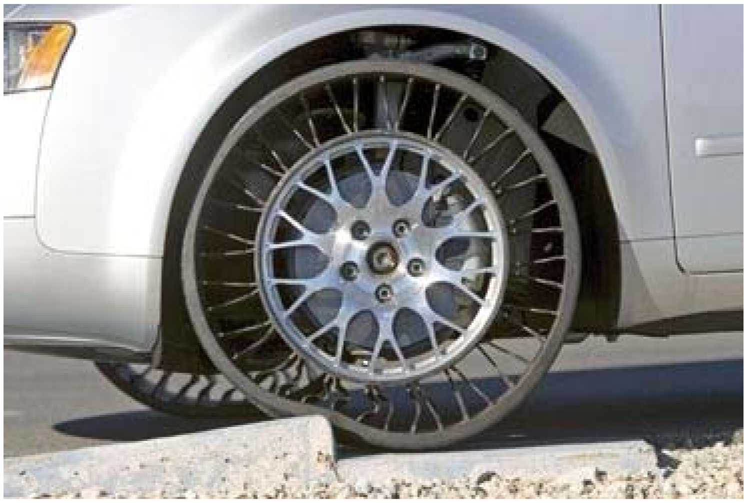


Figure 2.1 - Michelin TWEEL (Michelin Corporation, 2006)

Another rotating IWS is the wheels designed for NASA's Mars rover (National Aeronautics and Space Administration (NASA)). Figure 2.2 shows the mars rover wheels. The low temperature, lack of atmosphere, and also weight restrictions prevent the use of inflatable rubber tires in projects like the Mars rover. The designed mars rover wheel is about 10" in diameter and is cut out of a solid piece of aluminum. Radial flexible aluminum spokes are designed to isolate the sensitive electronic equipments on the rover from excitations caused by moving on the rough surface of Mars. The outer surface of the wheel is anodized or covered with a black coating for additional strength. The steering and traction mechanisms are placed inside the wheel and are protected from possible external debris with an open-cell foam called Solimide (Evonik Industries) which remains flexible at low temperatures. Although the Mars rover wheel is designed to isolate the rover from undesirable excitations, yet there is no data on the performance and mechanical response of the wheel officially published by NASA.

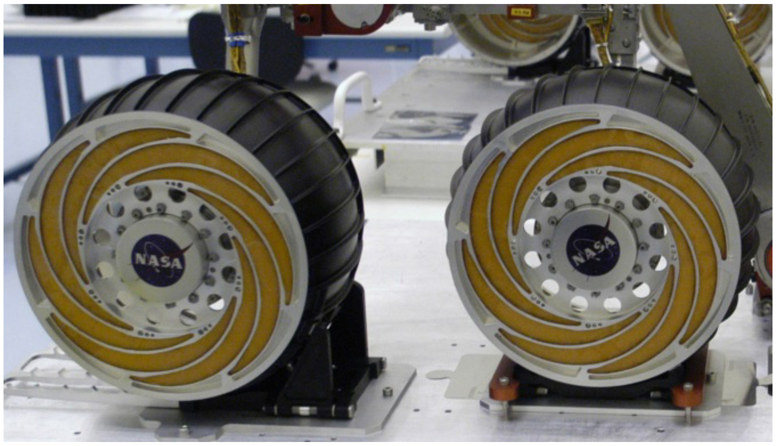


Figure 2.2 - NASA mars rover wheels (National Aeronautics and Space Administration (NASA))

In addition to the rotating isolating wheels, Michelin and Siemens have announced wheels with non-rotating suspension components placed inside them. Michelin active wheel (Michelin re-invents the wheel, 2008) and Siemens eCorner (Continental Automotive Systems Inc.) are shown in Figure 2.3 and Figure 2.4. The Michelin active wheel integrates an electrical motor, the power-train, the suspension system, and the braking system inside the wheel. Likewise, Siemens VDO's eCorner concept embeds an automobile's drive-train, shock absorbers, brakes, and steering, into its four wheels. The eCorner removes the need for the traditional engine architecture. Siemens declares "The transition from an internal combustion engine to an eCorner wheel hub motor will result in decreased emissions, increased energy efficiency and lower costs for consumers due to the elimination of hydraulic systems to maintain and service." (Continental Automotive Systems Inc.).

Although the Michelin and Siemens wheels embed different automobile subsystems, including the suspension system, inside its wheels, but the suspension system is still limited to isolation in the vertical direction. However, adding an additional degree of freedom to the suspension system of a high speed automobile requires more in-depth studies and considerations which are beyond the scope of this thesis.

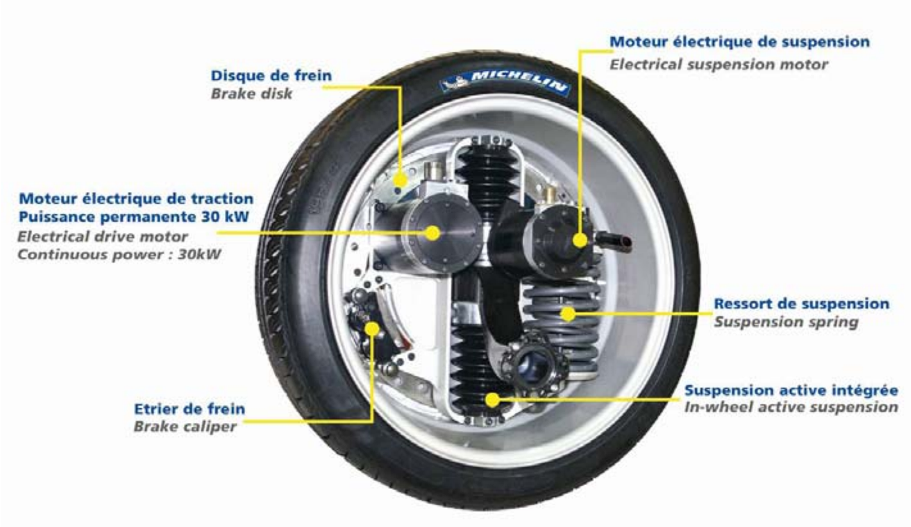

Moteur électrique de suspension
Electrical suspension motor
Disque de frein
Brake disk
MICHELIN
Moteur électrique de traction
Puissance permanente 30 kW
Electrical drive motor
Continuous power : 30kW
Ressort de suspension
Suspension spring
Suspension active intégrée
In-wheel active suspension
Etrier de frein
Brake caliper

Figure 2.3 - Michelin active wheel (Michelin re-invents the wheel, 2008)

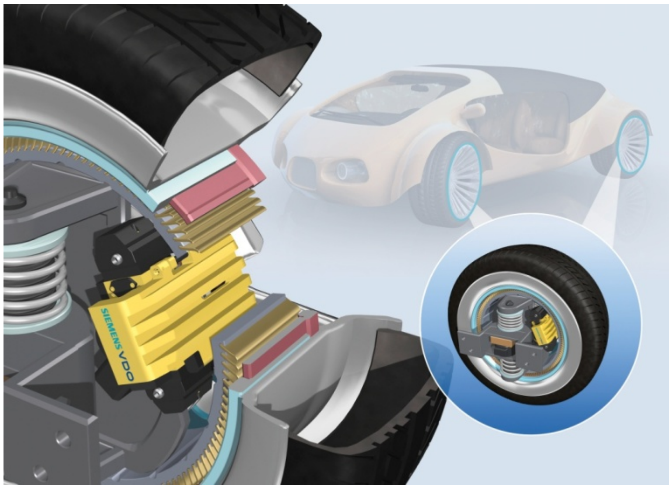


Figure 2.4 - Siemens' eCorner (Continental Automotive Systems Inc.)

2.2 Summary

This chapter has presented a survey of some of the researches for using the available space inside a vehicle's wheel. This place is used for placing useful parts such as suspension components, steering mechanisms, and even the power-train. However these researches are still in the primary stages, and there is no precise performance data or design guidelines presented. In the following chapters the practicality and performance of placing a suspension system inside a vehicle's wheel is studied in more detail.

Chapter 3
Investigation of the Dynamics of an In-Wheel Suspension

3.1 Introduction

Since the first introduction of suspension systems on automobiles in 1903 by Mors (Jain, et al., 2002), almost any vehicle is equipped with one. The suspension system is usually a spring-damper subsystem isolating the passenger from wheel excitations. Momentary loads from ground are absorbed and damped out by the suspension system. Tires are also part of the suspension system absorbing high-frequency excitations. Although, existing suspension systems can effectively reduce the transferred shocks in the vertical direction, yet the isolation of the shocks emerging from horizontal excitations has remained unsolved.

The in-wheel Suspension System (IWS) will address the technical deficiencies in existing suspension designs by providing more comfort. It can effectively isolate shocks and road disturbances in any direction in the plane of rotation. In addition to extra comfort, it provides added passenger safety, reduced power consumption and solid tire manufacturing opportunities in many applications. This chapter presents a dynamics study of an IWS which was designed for a wheelchair at the University of Waterloo (Khajepour, 2007). The design was originally based on static calculations. This chapter's objective is to evaluate the previously designed suspension for dynamics responses and investigate the effects of various parameters such as working speed, suspension stiffness, etc.

3.2 Model description

Figure 3.1 shows a schematic model of the wheelchair's IWS. The system consists of a planar mechanism with eight moving bodies:

- 1: Rim, which is pinned to the ground and has only one rotational degree of freedom.
- 2: Hub, which is connected to the rim via six links and has all of the three planar degrees of freedom with respect to the rim.
- 3...8: Six arms that are used to connect the rim and the hub together.

The suspension system's mechanical properties are shown in Table 3.1.

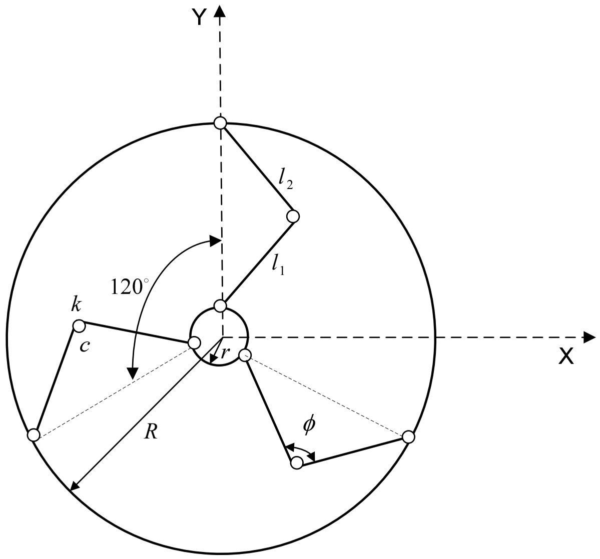


Figure 3.1 - Schematic model of RPS for manual wheelchairs

Table 3.1 - Parameter values (Khajepour, 2007)

Parameter	**Value**
k	20 N.m/rad
c	4 N.m.s/rad
Φ	90°
r	50 mm
R	240 mm
$l1,l2$	134.35 mm

The mechanism has four degrees of freedom, where one of the degrees of freedom (the Rim's rotation) is driven by a motor which specifies the operating speed of the wheel. A set of four independent variables is adequate to fully define the model; however it is more convenient to use ten dependent variables and six constraint equations. The ten variables are shown in Figure 3.2:

- X: The horizontal displacement of the center of the hub with respect to the ground
- Y: The vertical displacement of the center of the hub with respect to the ground
- Ψ: The angular displacement of the hub with respect to the rim

- *Ω:* The angle of rotation of the rim with respect to the ground
- $\theta_{1,3,5}$: The angles between the hub and the connecting arms
- $\theta_{2,4,6}$: The angles between the connecting arms

The six constraint equations can be found by the following vector equation:

$$\vec{OD} = \vec{OA} + \vec{AB} + \vec{BC} + \vec{BD} \tag{2.1}$$

Giving two scalar equations for each set of connecting arms, for three sets of connecting arms six constraint equations are defined. These constraint equations are actually imposed by the joints between the arms and the rim (*D* in Figure 3.2).

There is also another constraint equation which is imposed by the driver connected to the rim:

$$\Omega = \omega \times t \tag{2.2}$$

Where ω is the angular speed of the motor.

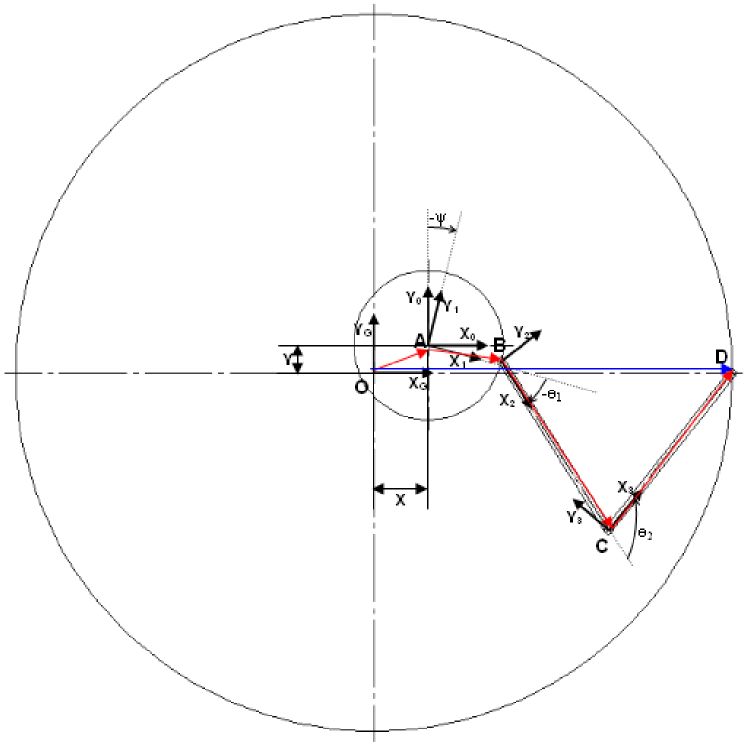


Figure 3.2 - Reference frames and variables

3.3 Deriving equations of motion

There are various methods that can be used to find the equations of motion for the IWS model; i.e. Newton-Euler’s Method, graph theory, and Lagrange's Method (Ginsberg, 1998). Using Newton-Euler’s method requires drawing free body diagrams and calculations at acceleration level. This method can be useful for deriving equations of motion by hand. Using the graph theory does not require free body diagrams, and has a straight forward algorithm to be implemented in a computer code; but the calculations are still at the acceleration level (Using linear graph theory and the principle of orthogonality to model multibody, multi-domain systems, 2008). Lagrange theory, not only does not require free body diagrams but also the calculation of accelerations is unnecessary, and the concept can be easily put into a computer code. Thus, Lagrange's method is used to find the equations of motion.

To use Lagrange's method, first the kinetic and potential energies have to be calculated. Knowing the potential and kinetic energies (*P.E* and *K.E*), constraint equations (*Φ*), and, the Jacobian (*Φq*), Lagrange method can be used to find equations of motion.

In matrix format:

$$[M]\left\{\ddot{q}\right\} = \left\{F(q,\dot{q},t)\right\} - \left[\Phi_q\right]^T \{\lambda\} \tag{2.3}$$

Where:

$$\{q\} = [X, Y, \Psi, \Omega, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$$

$$L = K.E. - P.E.$$

$$[M] = \left[\left[L_{\dot{q}}\right]^T\right]_q$$

$$\{F\} = \left[L_q\right]^T - \left[\left[L_{\dot{q}}\right]^T\right]_q \dot{q} + \{F_{external}\}$$

$\{\lambda\}$ is the reaction forces vector due to constraints

The procedure of finding the potential and kinetic energies is described in sections 3.3.1 to 3.3.6:

3.3.1 Coordinate systems

In order to simplify the vector operations a series of grounded or body fixed coordinate systems are defined (see Figure 3.2):

X_GY_G: global reference frame (grounded)

X_0Y_0: a reference frame at the center of the hub, translating with the hub but rotating with the rim (Ω)

X_1Y_1: a reference frame fixed to the hub at its center

X_2Y_2: a reference frame connected to $link_1$ one at point B

X_3Y_3: a reference frame connected to $link_2$ one at point C

Simple vector operations are used to transfer vectors for one reference frame to the other. A position vector defined in the i^{th} reference frame (${}^iA$) can be transferred to the j^{th} reference frame by a rotation matrix (${}^jR^i$), and a vector connecting the j^{th} reference frame's origin, O_j, to the i^{th} reference frame's, O_i, origin in the j^{th} reference frame (P_{ji}):

$$ {}^jA = {}^jR^i \times {}^iA + P_{ji} \tag{2.4}$$

$$\begin{cases} {}^jR^i = ({}^iR^j)^T \\ P^{ij} = -{}^iR^j \times P_{ji} \end{cases} \tag{2.5}$$

Equations (2.4) and (2.5) transfer the vector ${}^iA$ defined in the i^{th} reference frame to ${}^jA$ defined in the j^{th} reference frame.

3.3.2 Position vectors

In order to find the potential energy due to gravity it is mandatory to know the positions of the center of masses of the bodies with respect to the datum. The datum for gravity is set at the horizontal line passing through the rim’s center (X_G). Using equations (2.4) and (2.5) the center of mass of each body is first defined in its local reference frame then transferred to the global frame by a set of consequent transforms. For instance the center of mass of $link_1$ in the global frame would be:

$$ {}^GCM\_l_1 = ({}^GR^0 \times ({}^0R^1 \times ({}^1R^2 \times {}^2CM\_l_1 + P_{12}) + P_{01})) + P_{G0} \tag{2.6}$$

Where:

${}^2CM\_l_1$ = center of mass of *link*$_1$ in X_2Y_2 reference frame ($[\frac{l_1}{2}, 0, 0]^T$)

l_1 = length of *link*$_1$

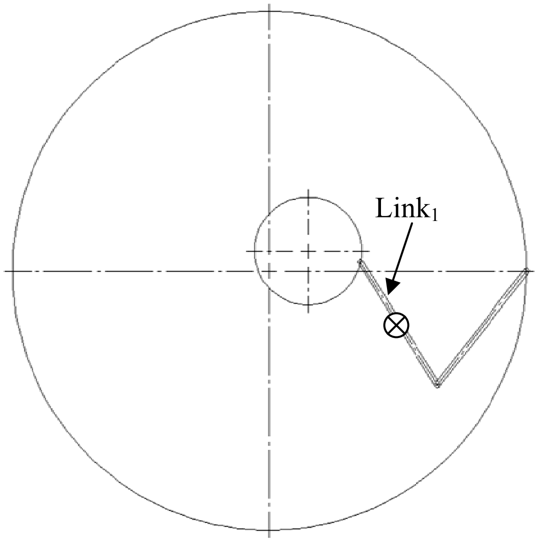


Figure 3.3 - Link1 and its center of mass

3.3.3 Velocities

Calculating the kinetic energy requires the speed of center of masses of the bodies with respect to the ground frame. Body fixed coordinates are used for the vector calculations, and all of the bodies are assumed to be rigid. Thus, relative velocities can be easily calculated for each body in its local coordinate. Afterwards, the velocities can be transformed from one frame to another and finally to the global frame. The concept is the same as that of the position vectors except that in each transformation the relative velocity of the origin and destination frames, V_{ji}, should be considered instead of their relative position (P_{ji}). I.e., a velocity vector, ${}^iu$, in i^{th} frame can be transformed to the j^{th} frame, ${}^ju$, by the following equation:

$$ {}^j\mathrm{u} = {}^j\mathrm{R}^i \times {}^i\mathrm{u} + \mathrm{V}_{ji} \quad (2.7)$$

Where:

$$ V_{ji} = (\vec{\omega}_j) \times \vec{P}_{ji} \quad (2.8)$$

ω_j=absolute angular velocity of j^{th} reference frame

For instance the absolute velocity of the center of mass of $link_1$ can be written in the ground reference frame as:

$$ {}^{G}u\_CM\_l^{1} = ({}^{G}R^{0}({}^{0}R^{1}(\,{}^{1}R^{2}\times{}^{2}u\_CM\_l_{1}\ +V_{12})+V_{01}))+V_{G0} \qquad (2.9)$$

Where:

${}^{2}u\_CM\_l_1$ = relative velocity of CM_l_1 in X_2Y_2 frame ($\overset{\circ}{\overrightarrow{{}^{2}CM\_l_{1}}}$) = $\overrightarrow{\dot{\theta}_{1}}\times\overrightarrow{{}^{2}CM\_l_{1}}$

$$V_{12} = (\overrightarrow{\dot{\Omega}}+\overrightarrow{\dot{\psi}})\times\overrightarrow{P_{12}}$$

$$V_{01} = (\overrightarrow{\dot{\Omega}})\times\overrightarrow{P_{01}}$$

$$V_{G0} = \left[\dot{X},\dot{Y},0\right]^{T}$$

The same approach is used to calculate the speed for the center of masses of all of the mechanism components.

3.3.4 Potential Energy

By knowing the center of mass positions in the global frame (section 3.3.2), the potential energy due to gravity ($P.E._g$) can be calculated (Datum is X_G):

$$P.E._{g} = \sum_{i=1}^{8}(({}^{G}CM_{i})\bullet\hat{Y}_{G})\times g\times M_{i} \qquad (2.10)$$

Where:

g = gravity acceleration = $9.81m/s^2$

M_i = mass of i^{th} body

The potential energy of the torsional springs ($P.E._k$) placed at the joints is equal to:

$$P.E._K = \frac{1}{2}\sum_{i=2,4,6} K \times \theta_i^2 \tag{2.11}$$

Where *K* is the torsional spring's rate.

And finally, the total potential energy is equal to:

$$P.E. \ = P.E._g + P.E._k \tag{2.12}$$

3.3.5 Kinetic Energy

The total kinetic energy is the sum of kinetic energies of the bodies:

$$K.E. = \sum_{i=1}^{8} K_i \tag{2.13}$$

$$K_i \ = \ \frac{1}{2}\left[M_i\left({}^G u\_CM_i\right)^T \times \ \left({}^G u\_CM_i\right) + I_i \ \left(\omega_i\right)^T \ \left(\omega_i\right)\right] \tag{2.14}$$

Where:

K_i = kinetic energy of i^{th} body.

I_i = mass moment of inertia for the i^{th} body about its center of mass

ω_i = absolute angular velocity of the i^{th} body = absolute angular velocity of the body's reference frame

3.3.6 Generalized Forces

Since gravity is already considered in the potential energy, if the system has no damping, the only external force would be the external load applied to the hub (along Y_G):

$$F_{external} = [0, F_y, 0, 0, 0, 0, 0, 0, 0, 0]^T \tag{2.15}$$

For a design with dampers located at the joints connecting the arms to each other:

$$F_{external} = [0, F_y, 0, 0, 0, C.\dot{\theta}_2, 0, C.\dot{\theta}_4, 0, C.\dot{\theta}_6]^T \tag{2.16}$$

Where *C* is the torsional damping coefficient.

3.4 Solving the Equations of Motion

After deriving the equations of motion and the constraint equations, the next step is to solve the Differential Algebraic Equations, DAEs (2.17). For this problem, there are ten independent variables, six geometric constraint equations, one driver constraint equation, and, seven Lagrange multipliers, giving a total of 17 unknowns and 17 DAEs. The DAEs are solved by converting them to Ordinary Differential Equations (ODEs) by differentiating the constraint equations and solving for *{λ}:*

$$\begin{cases} [M]\left\{\ddot{q}\right\} = \left\{F(q,\dot{q},t)\right\} - \left[\Phi_q\right]^T.\{\lambda\} \\ [\Phi] = 0 \end{cases} \tag{2.17}$$

$$[\ddot{\Phi}] = [\Phi]_q\left\{\ddot{q}\right\} + [\Phi]_{tt} + 2[\Phi]_{qt}\left\{\dot{q}\right\} + ([\Phi]_q\left\{\dot{q}\right\})_q\left\{\dot{q}\right\} = 0$$

$$\Rightarrow [\Phi]_q\left\{\ddot{q}\right\} = -([\Phi]_{tt} + 2[\Phi]_{qt}\left\{\dot{q}\right\} + ([\Phi]_q\left\{\dot{q}\right\})_q\left\{\dot{q}\right\}) = \{\gamma\} \tag{2.18}$$

$$\therefore \begin{cases} [M]\left\{\ddot{q}\right\} = \left\{F(q,\dot{q},t)\right\} - \left[\Phi_q\right]^T\{\lambda\} \\ [\Phi]_q\left\{\ddot{q}\right\} = \{\gamma\} \end{cases} \quad or \quad \begin{bmatrix} [M] & [\Phi]_q^T \\ [\Phi]_q & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{q}\} \\ \{\lambda\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{\gamma\} \end{Bmatrix}$$

Since [Φ]=0 is not necessarily satisfied anymore, solving ODEs instead of DAEs may result in errors at joint locations. However the simulation needs to be run only for a few seconds (until the steady state response) and the error can be minimized by using small time steps.

3.5 Simulation Results

After converting the DAEs to ODEs, the ODEs can be solved using MATLAB's standard ODE solvers such as ODE45 (MathWorks, Inc., 2006). Figures 3.4–3.6 show MATLAB results for parameters *X*, *Y*, and θ_2 of the un-damped mechanism for a two second simulation. The initial conditions are shown in equations (2.19) and (2.20).The results are also compared with an ADAMS model to verify the equations of motion and results.

$$\{q_0\} = [0,0,0,0,-\pi/4,\pi/2,2.\pi/3-\pi/4,\pi/2,4.\pi/3-\pi/4,\pi/2]^T \tag{2.19}$$

$$\left\{\dot{q}_0\right\} = [0,0,0,2.\pi,0,0,0,0,0,0]^T \tag{2.20}$$

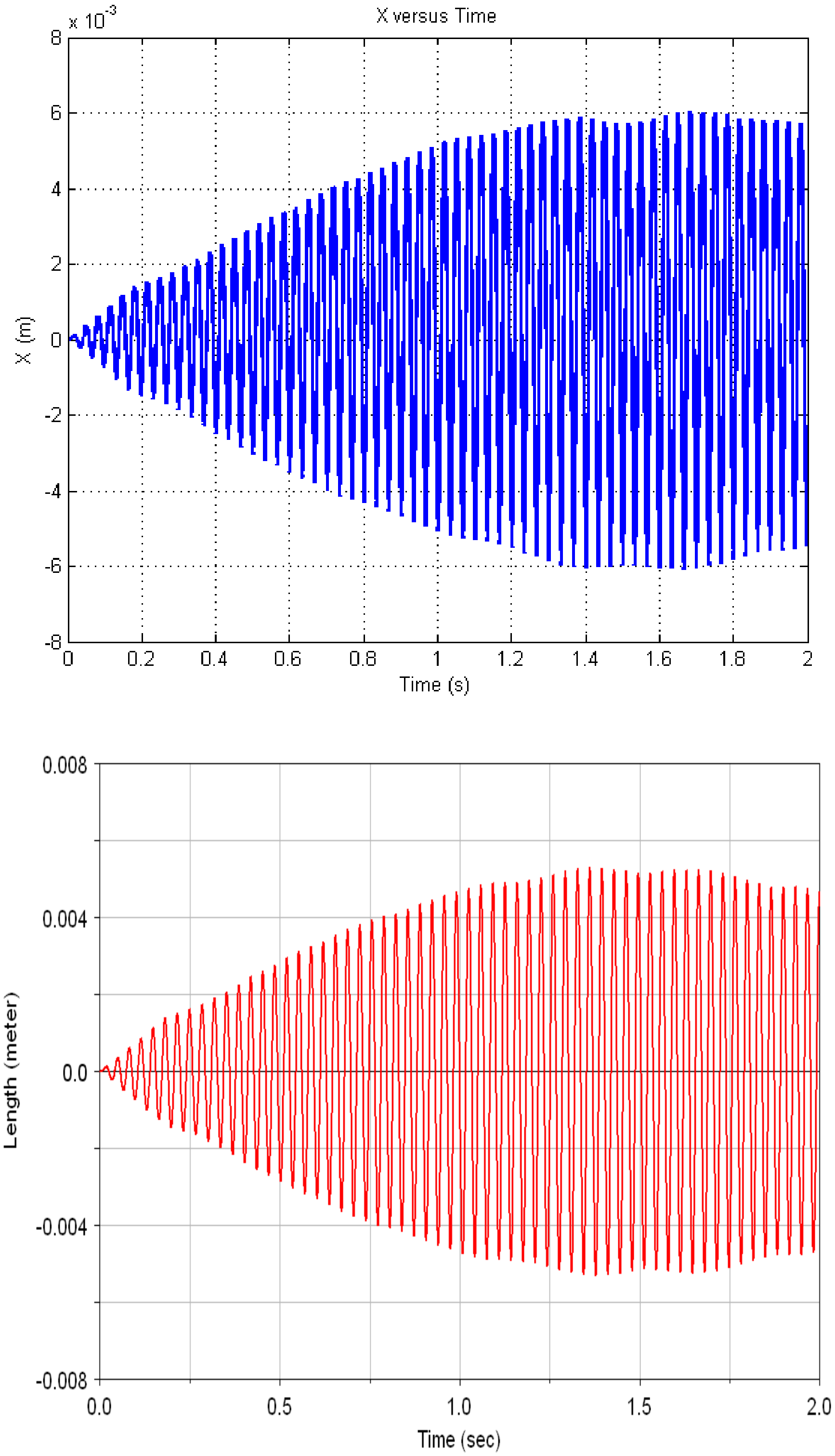


Figure 3.4 – Horizontal displacement results from MATLAB (top) and ADAMS (bottom)

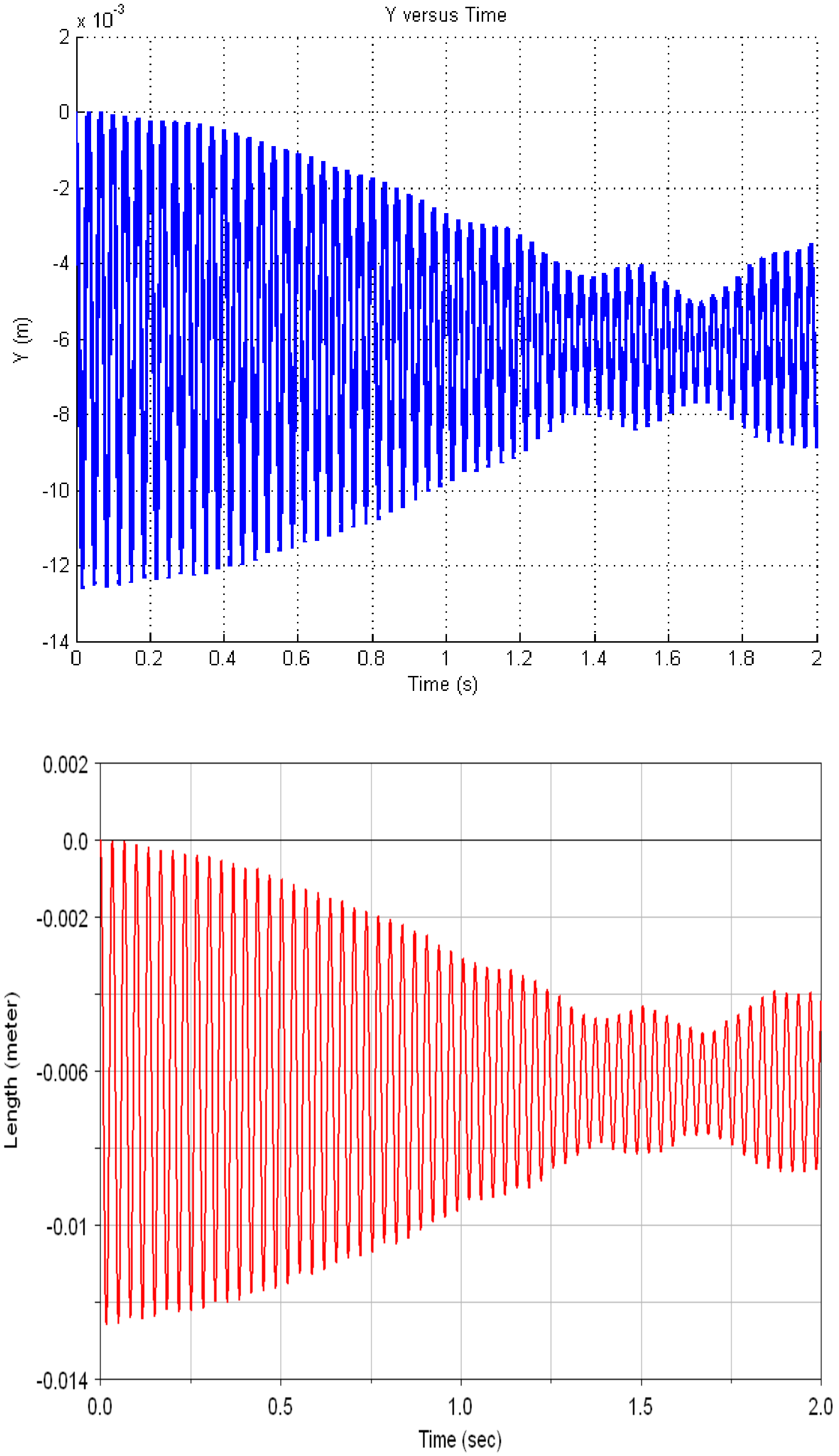


Figure 3.5 - Vertical displacement results from MATLAB (top) and ADAMS (bottom)

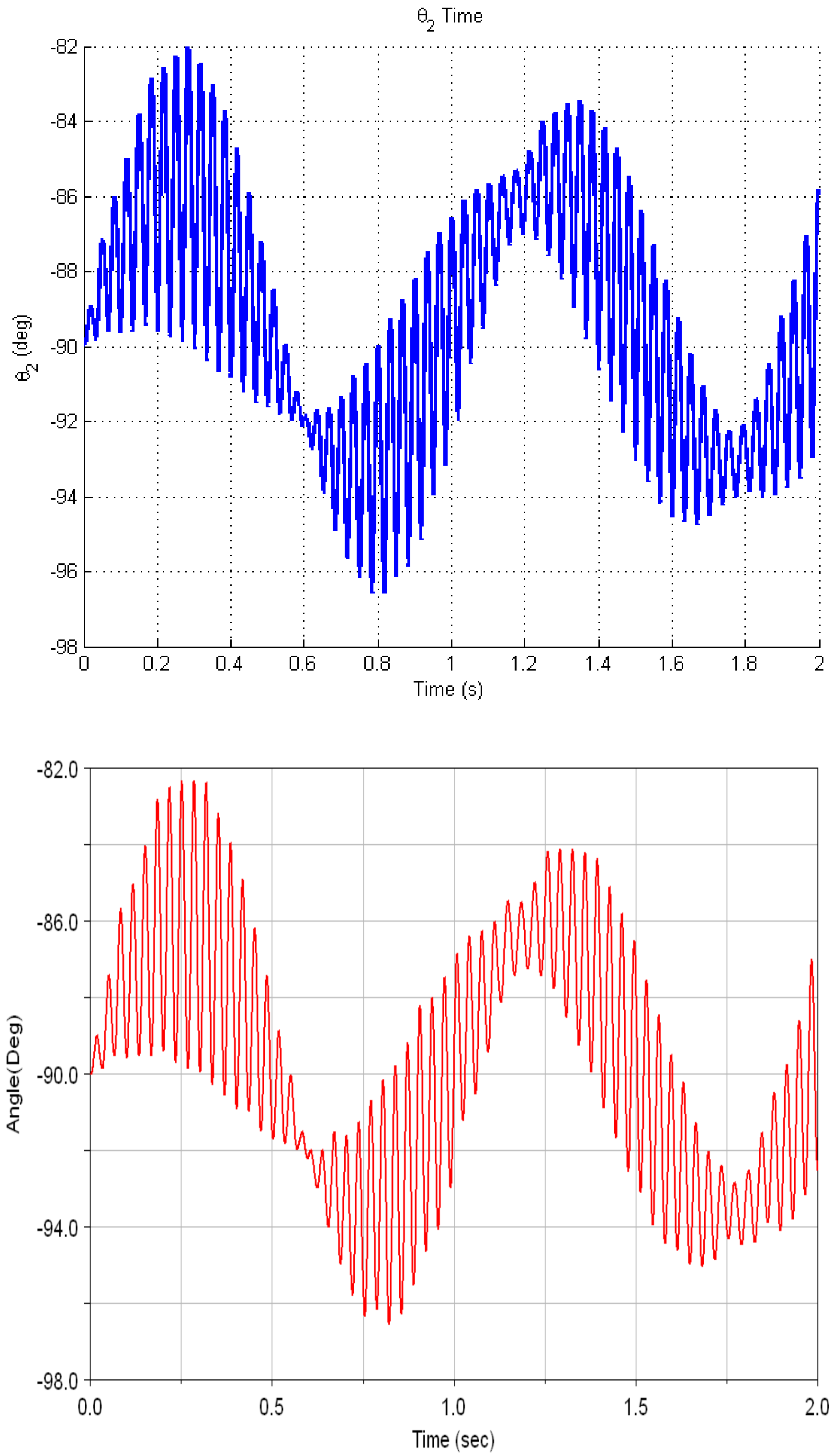


Figure 3.6 – Angular displacement results from MATLAB (top) and ADAMS (bottom)

The dynamics model can be further investigated by adding damping to the system (*C=4 N.m.s/rad*). Figure 3.7 shows the simulation results for various vertical loads. The rotating speed is set to 600RPM for the rim which is equal to a linear speed of 54KPH. The transient response to the applied vertical load is visible at the beginning of the simulation. As shown in the graphs the vertical displacement increases approximately linearly with the applied load. The stiffness fluctuation effects are more visible at higher loads, but the amplitude of the fluctuation in vertical and horizontal directions remains within a fraction of millimeter and therefore, negligible.

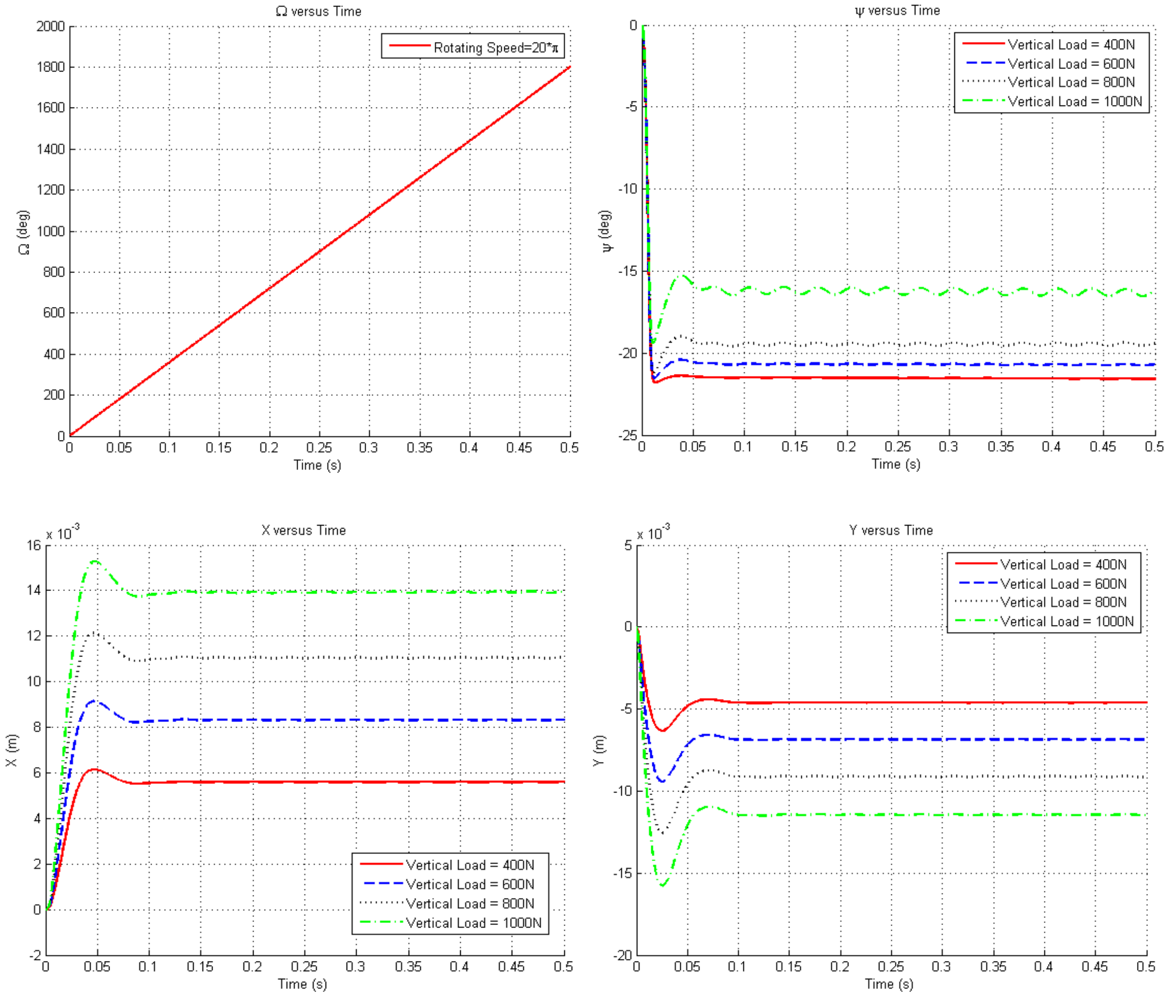


Figure 3.7 - Simulation results for various vertical loads

The influence of the working speed is shown in Figure 3.8. The results show that the suspension becomes stiffer as the working speed is increased. This is mainly due to the centrifugal forces of the rotating arms. When a vertical load is applied to the hub, the vertical displacement ruins the mechanism's symmetry resulting in reacting forces trying to return the system to the initial symmetric configuration. As it is shown in the graphs, for rotating speeds higher than 600RPM (54KPH) the displacement becomes less than 45% of the initial displacement or the suspension becomes more than two times stiffer. This phenomenon can be considered as a speed limit for the effectiveness of the suspension system.

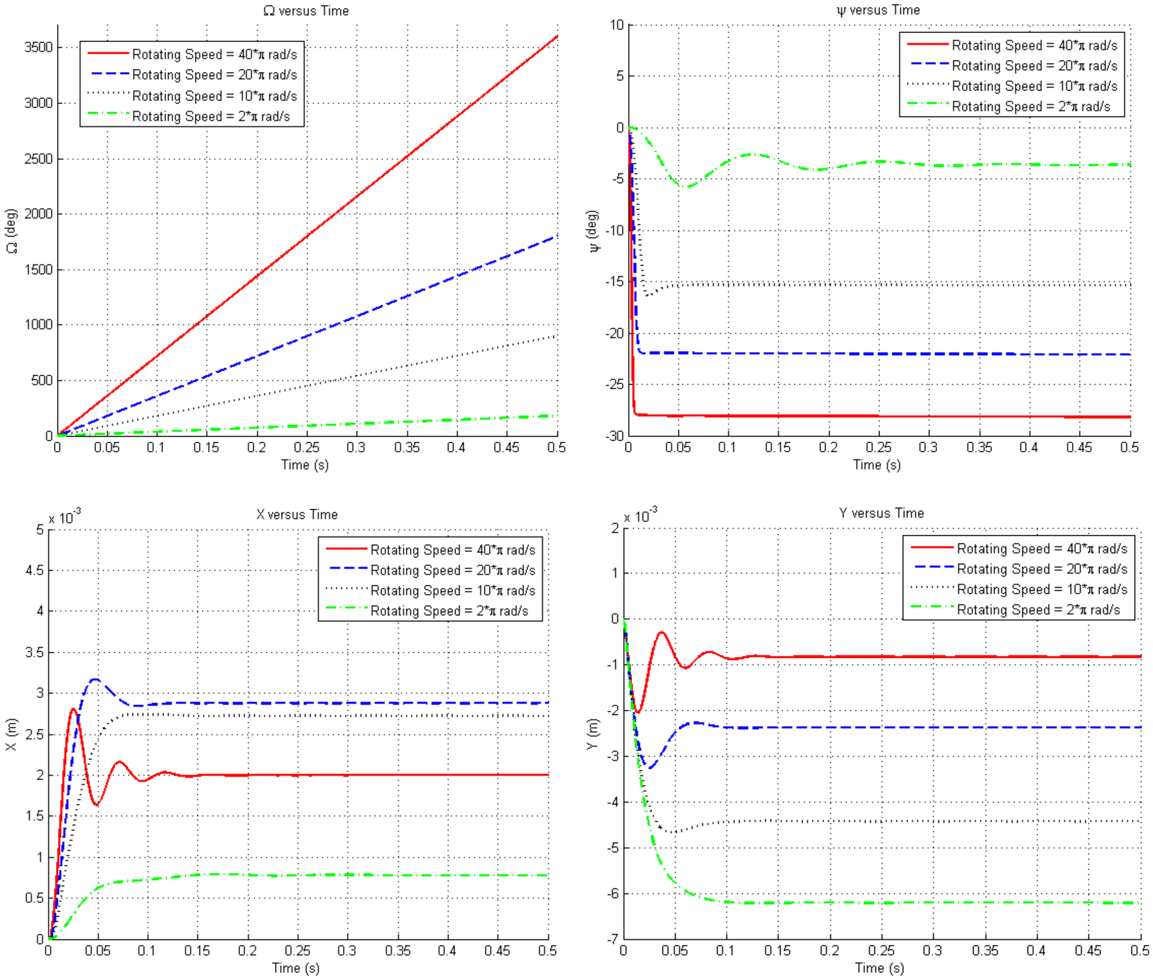


Figure 3.8 - Simulation results for various rotating speeds

Figure 3.9 displays the simulation results for different masses for suspension arms. It can be seen that heavier links result in more vertical displacement which is due to the effect of gravity. Moreover, it is demonstrated that the suspension stiffness fluctuations are more visible for heavier arms. Thus, using lighter arm not only increase the useful wheel travel by minimizing the static displacement but also helps to eliminate the undesirable stiffness fluctuations.

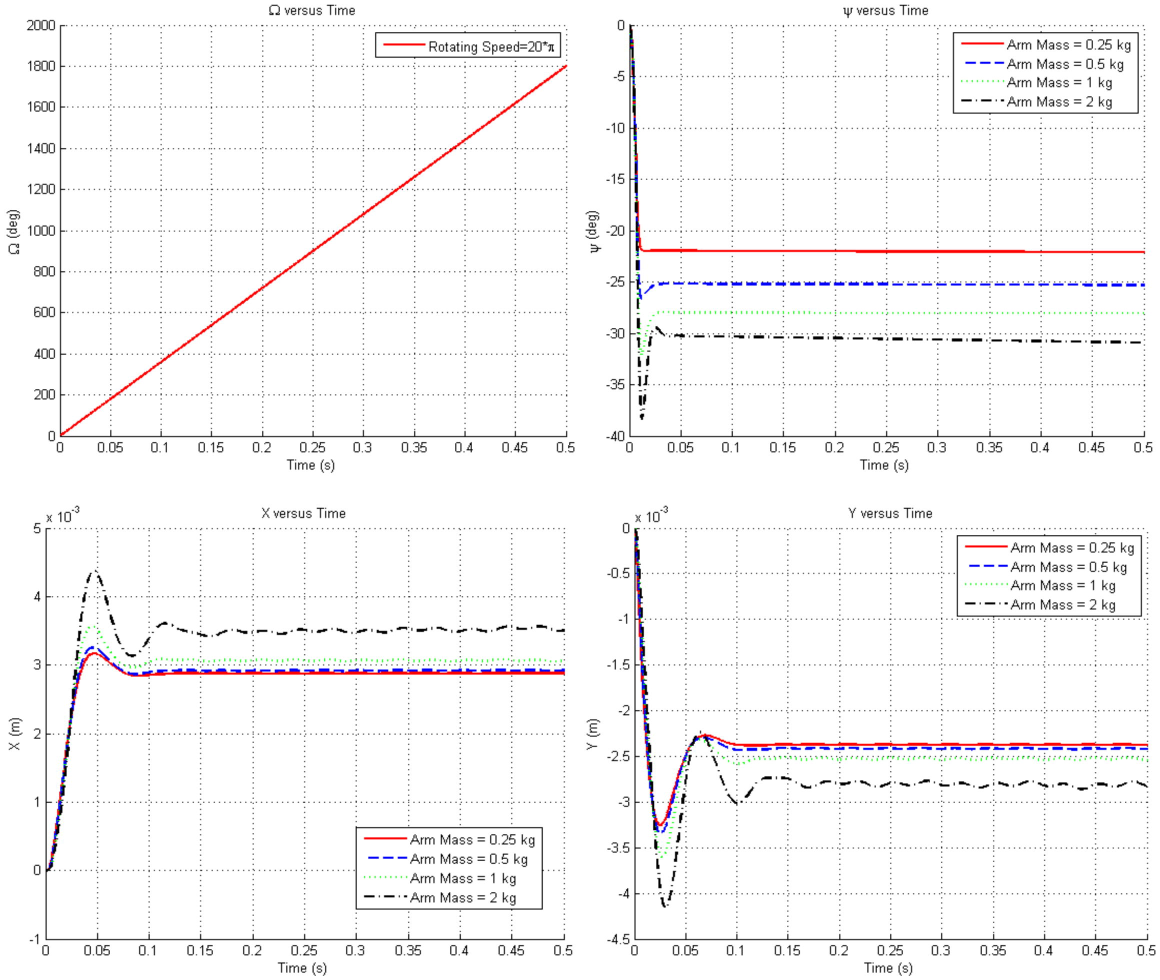


Figure 3.9 - Simulation results for various arm masses

3.6 Summary

This chapter has presented the dynamics investigation of a mechanism based rotating in-wheel suspension system. The dynamics response simulations exhibited the effects of various parameters on the rotating suspension. The results demonstrated the linear behavior of the suspension stiffness, and also verified the insignificance of stiffness fluctuations when the mechanism arms are perpendicular to each other. Furthermore, the results showed the dependence of a rotating suspension system's stiffness on its operating speed: at higher speeds the suspension effectiveness fades due to internal forces resisting the unsymmetrical configuration of the mechanism imposed the external load. Moreover, it was seen that the suspension performance is improved by reducing the mass of the rigid suspension links.

Chapter 4
Compliant In-Wheel Suspension

4.1 Introduction

Although the rigid, mechanism based, in-wheel suspension demonstrates the feasibility of embedding a vehicle's suspension system inside its wheel, but the design's complexity, comparatively high weight of its components, and moreover the wheel travel restriction due to the spacious rigid members, demonstrate the potential to refine the design. The in-wheel suspension can be improved by replacing the rigid mechanism arms with flexible members. The flexible members connect the hub of the wheel to its rim, acting not only as the suspension springs, but also as the connecting arms. While using flexible members, reduces the number of components and their weight, and also provides more space for wheel travel; designing the flexible members in order to provide specified stiffness, wheel travel, and required life span is a design challenge which is addressed in this chapter. First Castigliano's energy method is used to design and evaluate spring with analytically defined shapes, and then the design optimization is improved by using an external FEA solver (ANSYS). Finally, a MATLAB code is developed to perform the spring shape optimization in MATLAB and ANSYS. The software provides a Graphical User Interface (GUI) in MATLAB allowing the user to input design specifications and run the optimization without dealing with the complexity of using the FEA software.

Although metal springs are widely used in automotive and other industries, but the spring shapes are usually limited to standard coils or strips by manufacturing processes. However the superior manufacturing capabilities, diversity, physical and mechanical properties of polymers and composite materials over metals have encouraged designers to replace metal parts with non-metal components, allowing for further freedom in design and shape optimization. Using non-metal flexible materials such as composites and polymers have become an indispensable part of the automotive industry. Several researches are devoted to the use of non-metal materials in automobiles. Some of these studies focus on the design of non-metal springs. Breadmore (Composite structures for automobiles, 1986), (The potential for composites in structural automotive applications, 1986) studied the application of composite structures for automobiles. Moris (Composite integrated rear suspension, 1986) focused on implementing composite materials in the rear suspension system. Daugherty (Composite leaf springs in heavy truck applications, 1981) considered the use of non-metal leaf spring in heavy trucks. Yu and Kim (Double tapered FRP beam for automotive suspension leaf spring, 1988) optimized a double tapered leaf spring for automotive suspension. Corvi (A preliminary approach to composite beam design using FEM analysis, 1990) used FEM methods in composite leaf spring design. In this section the design and shape optimization of springs is studied. Although the same methods might be used for optimization of metal and non-metal springs, but manufacturing an optimized metal spring might not be practically feasible.

4.2 Designing Flexible Members Using Castiglione's Method

In this section an analytical energy approach is used to design and optimized spring shapes.

4.2.1 Castigliano's Energy Method

The strain energy of a beam subjected to multiple loads can be calculated in terms of the applied loads and the resulting deflections (Beer, et al., 1985).

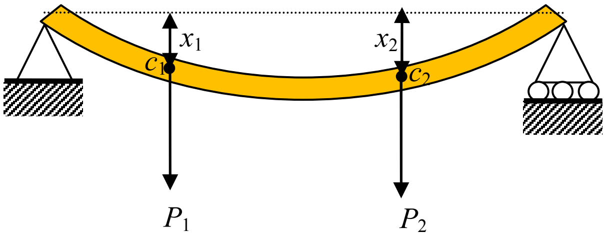


Figure 4.1 - Schematic drawing of a beam subject to concentrated loads

Consider two concentrated forces P_1 and P_2 applied to an elastic beam *AB*. The strain energy of the beam is equal to the work of the loads as they are slowly applied at C_1 and C_2 (Figure 4.1). The elastic deflections x_1 and x_2 can be expressed in terms of P_1 and P_2:

$$\begin{array}{ll} x_{11} = \alpha_{11} \times P_1 & x_{21} = \alpha_{21} \times P_1 \\ x_{12} = \alpha_{12} \times P_2 & x_{22} = \alpha_{22} \times P_2 \end{array} \tag{3.1}$$

Where:

x_{ij} = the deflection at C_i due to P_j

α_{ij} = *influence coefficients*, representing the deflection at C_i when a unit force is applied at C_j

Applying the principle of superposition, the deflection x_1 and x_2 can be expressed by the following equation:

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} \tag{3.2}$$

The work done by P_1 and P_2 depends on the order that the forces are applied, if P_1 is applied first, the strain energy can be calculated from equation(3.3); while, if P_2 is applied first the strain energy would be equal equation(3.4) .

$$U = \frac{1}{2}(\alpha_{11}P_1^2 + 2\alpha_{12}P_1P_2 + \alpha_{22}P_2{}^2) \tag{3.3}$$

$$U = \frac{1}{2}(\alpha_{11}P_1^2 + 2\alpha_{21}P_1P_2 + \alpha_{22}P_2{}^2) \tag{3.4}$$

However, as the two values should be the same, it can be concluded that:

$$\alpha_{12} = \alpha_{21} \tag{3.5}$$

Differentiating the strain energy by x_1 and x_2 yield to Castigliano's theorem:

$$x_1 = \frac{\partial U}{\partial P_1} \qquad x_2 = \frac{\partial U}{\partial P_2} \tag{3.6}$$

More generally, the deflection x_i on an elastic structure subject to a series of concentrated loads P_1, P_2, ..., P_n, at the point of application of P_i can be expressed as the partial derivative of the strain energy with respect to the load P_i.

$$x_i = \frac{\partial U}{\partial P_i} \tag{3.7}$$

Although the theory seems to be limited to calculating deflection only at points where a load is applied, but the deflection can be calculated at any desired point by applying a dummy force at that point and then setting the force's magnitude to zero at the end of the calculations.

In the special case of an elastic beam subject to pure bending, the strain energy can be calculated using equation(3.8). The deflection can also be calculated using Castigliano's theorem(3.9).

$$U = \int_0^L \frac{M^2}{2EI} dx \tag{3.8}$$

Where:

M = bending moment

E = Young's modulus

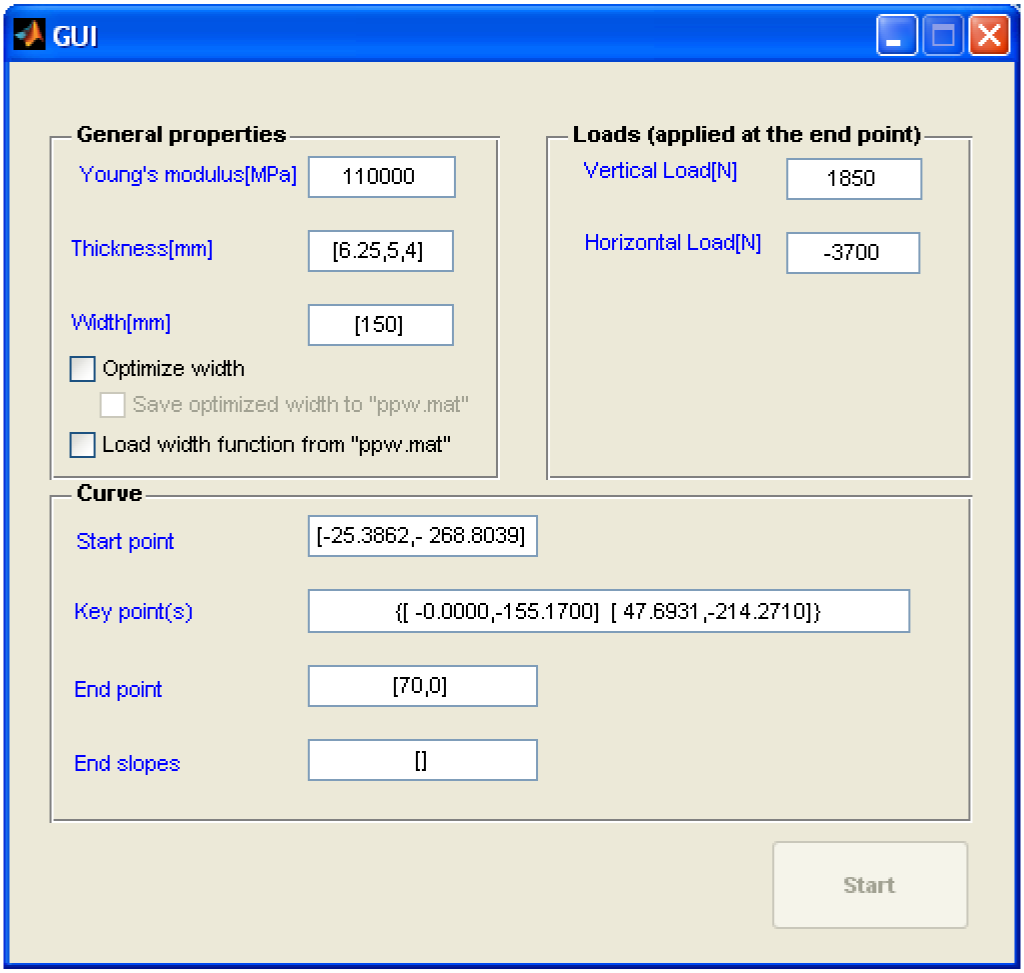
I = cross section inertia

$$x_j = \frac{\partial U}{\partial P_j} = \int_0^L \frac{M}{EI} \frac{\partial M}{\partial P_j} dx \tag{3.9}$$

4.2.2 Applying Castigliano's Theorem to Spring Design

A general shaped planar spring can be considered as a beam with a constant or variable cross section subject to an in-plane force vector. As springs are usually designed within the elastic range of the material, Castigliano’s theorem can be used to simulate their behavior. However, it should be noticed that, in addition to the elastic behavior of the spring, the pure bending assumption requires the spring to be thin otherwise shear forces should be considered as well. Moreover, the solution may not be precise for large deformations as non-linear geometry effects are neglected.

The implementation of Castigliano's theorem requires calculation of the spring's strain energy. Using equation (3.8) to calculate the spring's strain energy requires the beam's bending moment (*M*), and cross-section properties (*I*) to be defined as a function the beam's length (*x*). To do so, a MATLAB code is developed which inputs the spring as a series of defining key points. Figure 4.2 shows a screen capture of the code's GUI. The spring's geometry is defined by a starting key point (Start point), followed by a series of key points (Key point(s)), and finally a key point at the end of the spring (End point). The code passes a cubic Spline through the key points. The sloped at the beginning and the end of the spring can be defined too (End slopes). Figure 4.3 shows a schematic view of a sample spring. The cross section properties, thickness and width, can be defined as a constant value for the entire length of the spring or with different values at each key point which will be linearly interpolated for the rest of the spring’s length.



Young's modulus[MPa] 110000

Thickness[mm] [6.25,5,4]

Width[mm] [150]

□ Optimize width

□ Save optimized width to "ppw.mat"

□ Load width function from "ppw.mat"

Vertical Load[N] 1850

Horizontal Load[N] -3700

Start point [-25.3862,- 268.8039]

Key point(s) {[-0.0000,-155.1700] [47.6931,-214.2710]}

End point [70,0]

End slopes []

Figure 4.2 - Castigliano spring design GUI

The program then uses the analytical polynomial definition of the spring to calculate the bending moment, and section properties along the beam's length. Accordingly equation (3.8) is used to integrate the strain energy along the spring. The integrating points for a sample spring are shown in Figure 4.3. Finally the strain energy is partially differentiated to find the deflection at the spring's end point. The stress distribution can also be calculated at each section of the beam, using the following equation:

$$\sigma = \frac{Mc}{EI} \tag{3.10}$$

Where:

σ = the maximum stress at each section

c = half of the section height at the desired point ($\frac{t}{2}$)

As the stress at each point of the spring is linearly related to the section width at that point, the program can optimize the section width along the spring to achieve a spring with a relatively constant stress distribution.

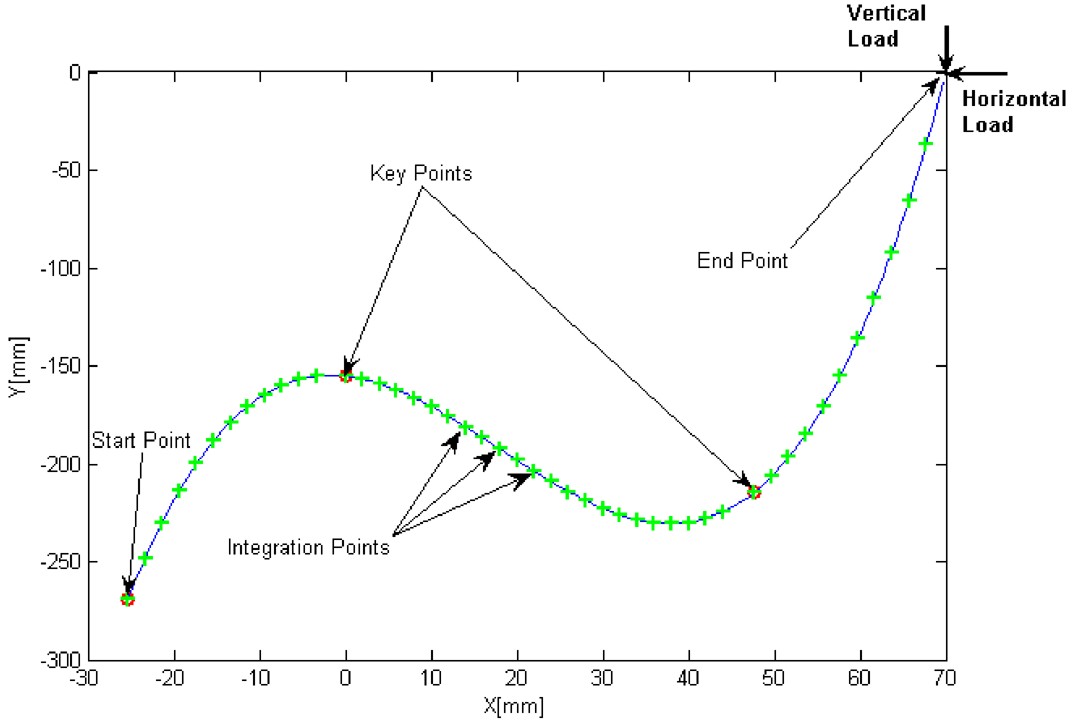


Figure 4.3 - Schematic view of a spring

MATLAB's optimization functions such as *fminmax* can be used to optimize the shape of the Castigliano spring to design a spring with a specific stiffness. Figure 4.4 shows a sample spring inside a wheel's rim. The graph on the right shows the spring's equivalent stress. A simple proof of concept for a non-rotating in-wheel suspension system is designed using the code (Figure 4.5). The design consists of four pivot joint springs and two cantilever springs. Four pivot joint springs are used instead of two to achieve a symmetric design and also to increase the capability to handle out of plane forces. The cantilever springs are added to the design to boost the horizontal stiffness of the design. Although, the Castigliano method can be used to design and optimize springs for the IWS, but analytical integration and solutions takes a long time in comparison to FEA methods. Moreover, the model is constrained to analytically shaped springs with small deformation; however one of the objectives of a suspension spring is to provide as much as wheel travel as possible. To overcome these restrictions, a more advanced spring optimization tool developed in the next section.

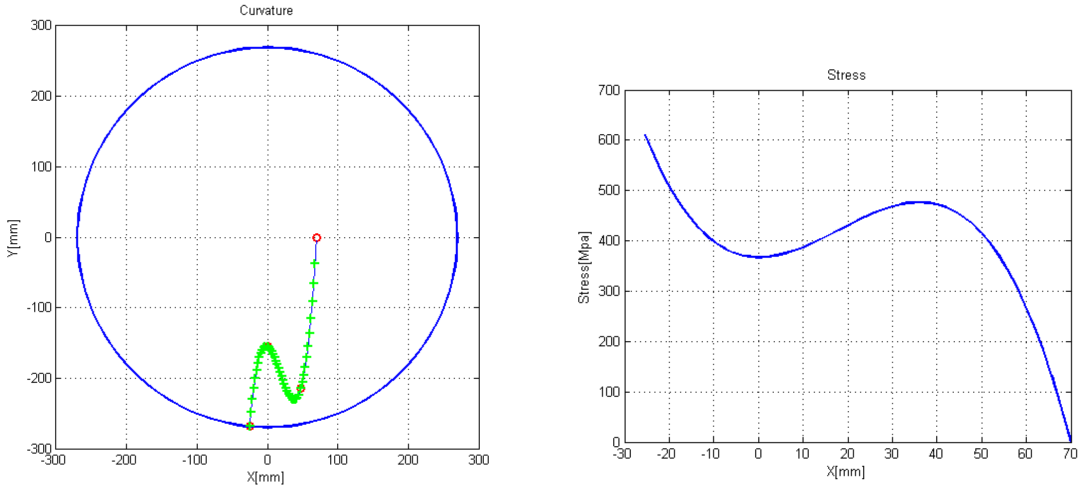


Figure 4.4 - Sample suspension spring and its stress levels

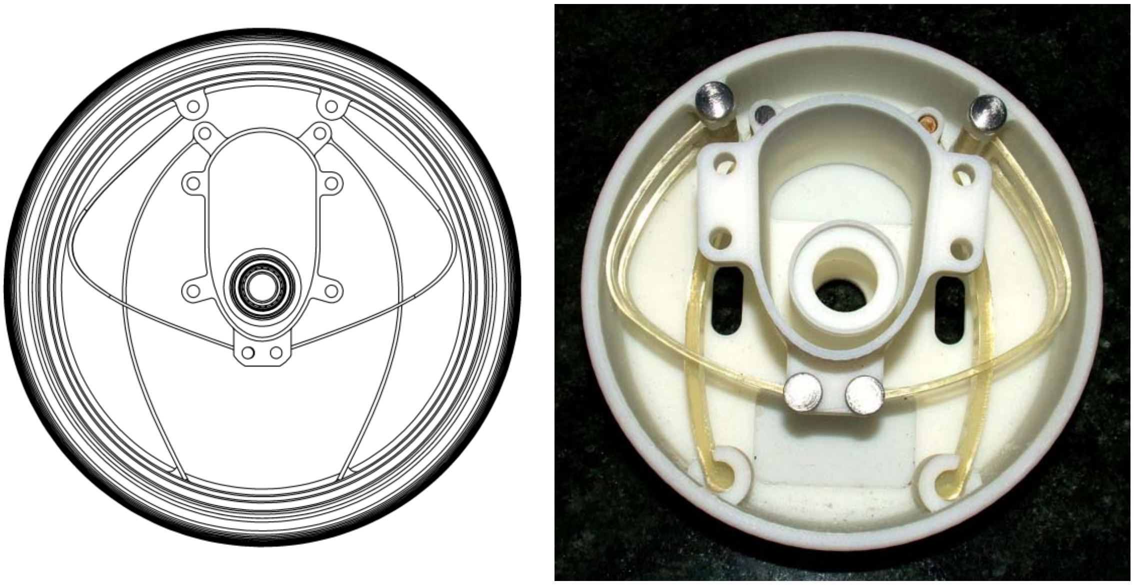


Figure 4.5 - Proof of concept design

4.3 Optimizing Flexible Members Using MATLAB and ANSYS

One approach to design an in-wheel suspension system is to fill the space between the wheel's hub and rim with a series of identical springs (Figure 4.6) that rotate with the wheel and provide the desired stiffness between hub and rim. The springs should be designed to allow for the maximum possible wheel travel while keeping the structural stress within the spring material's limits. A hard stopper mechanism should be also placed in the wheel physically confining the wheel travel, and avoiding any damages to the springs in unexpected overload conditions. Another challenge in designing such a suspension is the possibility of fluctuations in the suspension's stiffness as the wheel rotates. The rotation of the wheel, and consequently the embedded suspension system, changes the spatial configuration of the spring at different wheel angles which can result in stiffness fluctuations. Such fluctuations will make the isolated body to oscillate when the wheel is rotating on a flat smooth surface.

This section improves the spring design and optimization method introduced in section 4.2 to a practical tool for designing in-wheel suspension systems. This is made possible by using a more sophisticated model and FEA solver. The spring geometry is still defined by key points; however there is no need for the spring shape to be analytically defined. Moreover, using a powerful FEA tool allows for considering large deformations and non-linear geometry options combined with more accurate results and lower simulation times. Furthermore the non gradient optimization method allows for consideration of structural stress in the optimization.

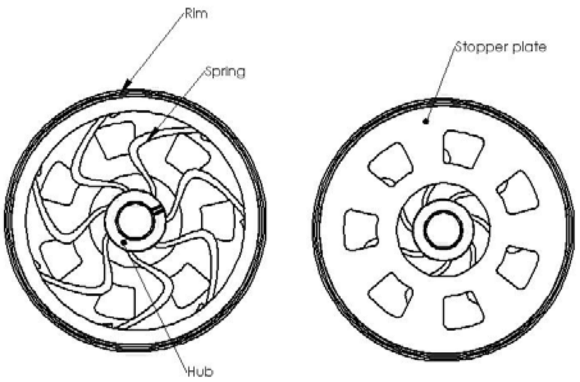


Figure 4.6 -In-wheel suspension system

4.3.1 Model Parameterization

The first step in an optimization problem is to fully define the design with a set of design variables. The next step is defining a cost function to evaluate the design variables for each of the optimization iterations. Next, the design variables are modified depending on the cost function value and the optimization method to obtain the next set of design variables. This procedure is repeated until either an acceptable cost value is achieved or the number of iterations exceeds a specified limit.

The suspension spring design discussed in this section is the shape optimization of discrete identical flexible members. Therefore, the spring topology should be defined before starting the optimization. Although there might be various topologies to choose from; it is suggested to start with a simple model. For instance, each of the identical suspension springs can be modeled as a spline with three control points and two end slope values (Figure 4.7). The section height and width of the spring are also defined at some key locations and change linearly along the spline's length.

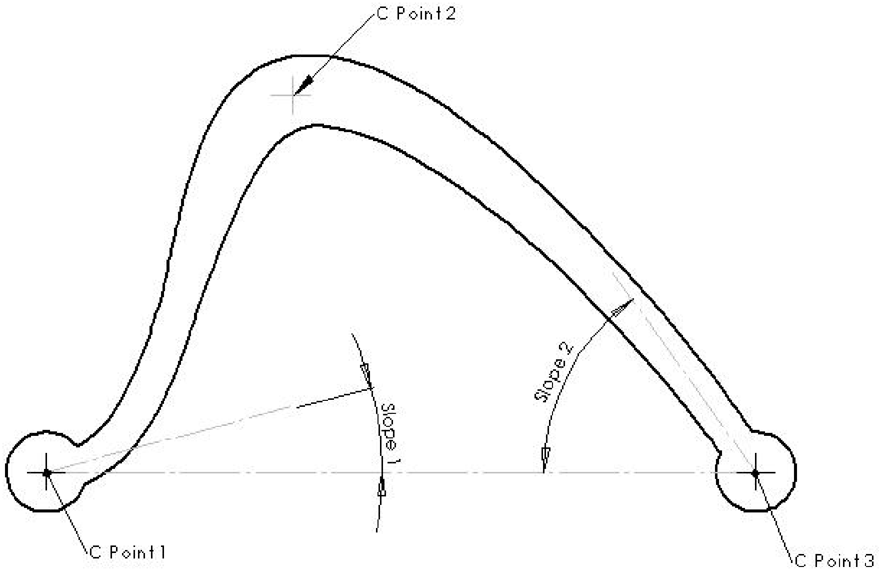


Figure 4.7 - Sample parameterization of a spline spring

The same approach can be used to define other spring shapes as well. Figure 4.8 shows two different designs that are created using the same key point concept. Basically, any two dimensional shape can be modeled with a series of defining key points; however a more complicated shape would require more defining key points, consequently more design variables and longer simulation time. General inputs to the FEA solver, which are discussed in the section 4.3.4 are a series of key points, material properties, and loading options that are used to create the FEA model.

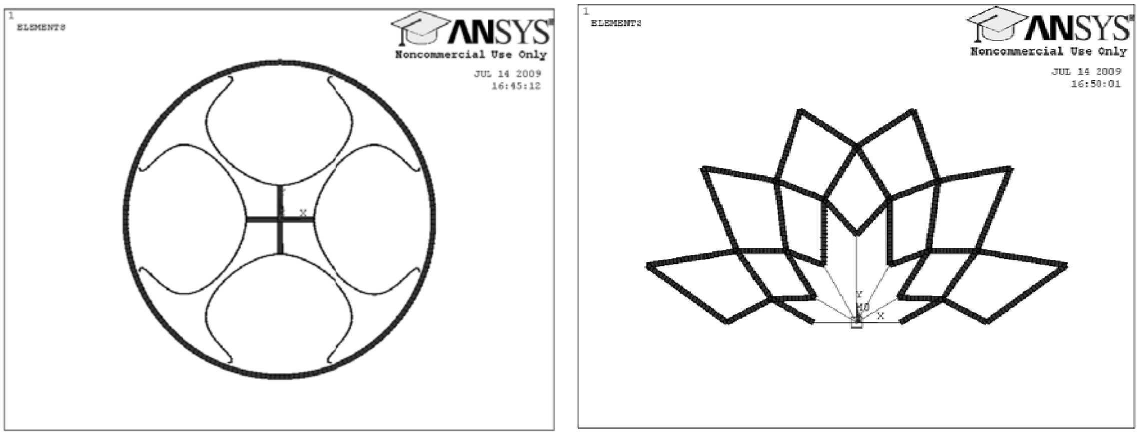


Figure 4.8 - Different shapes created using key points

4.3.2 Cost Function

After defining the necessary design parameters to define the shape of the suspension spring; the performance quality of the spring should be evaluated in each of the optimization iterations. The function that evaluates the spring performance (i.e. comparing the spring to the specified stiffness and stress values) is called the cost function.

The cost function consists of two parts; one evaluating the stiffness of the spring and the other one invesigatinging the stress values. The inputs to the cost function are the calculate stiffness and stress values from the FEA solver. The stiffness cost value function (3.11) is the square of the difference between the calculated stiffness value (K_i) and the desired stiffness (K_d). As the suspension's stiffness should not fluctuate when the wheel rotates, the stiffness may be calculated in more than one orientations of the wheel ($i=1,\ 2\ldots\ n$). In this case the stiffness cost function would be equal the sum of the square of differences for each orientation.

$$Stiffness\,\cos t - value\ funtion = \left[\sum (K_i - K_d)^2\right] \tag{3.11}$$

The behavior of the stiffness cost function is shown in Figure 4.9. The cost increases exponentially when the calculated stiffness is either higher or lower than the desired value.

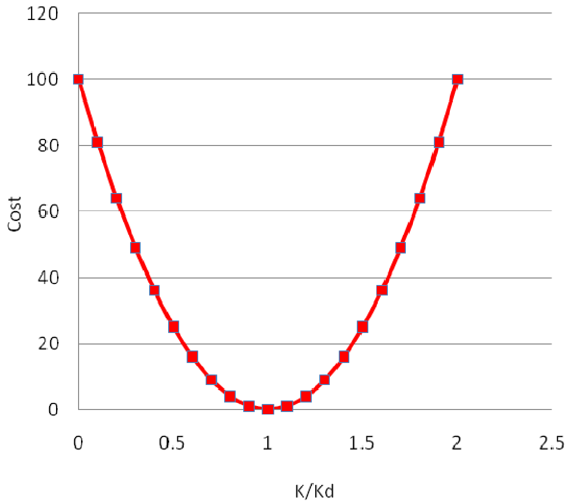


Figure 4.9 - Stiffness cost function behavior

The stress cost function assesses the maximum stress (S_{max}) calculated by the FEA solver. A combination of logarithmic and polynomial functions is used to achieve different function behaviors for stress values over or below the design stress (S_d).

$$Stress\cos t - value\ function = \left[(\frac{S_{\max}}{S_d} - 1) \times \log(\frac{S_{\max}}{S_d} + 2) \right]^2 \tag{3.12}$$

As it is shown in Figure 4.10, the stress cost increases exponentially when the stress levels are over the design stress; however for stress levels below the design stress the function increases slowly. This behavior is desirable as designs with stress levels below the desired stress are acceptable. A perfectly designed spring is a spring which has the same level of stress distributed evenly over the spring's material. The small increase in the stress cost function for stresses lower than the design stress helps to minimize the unbeneficial material not contributing in storing energy.

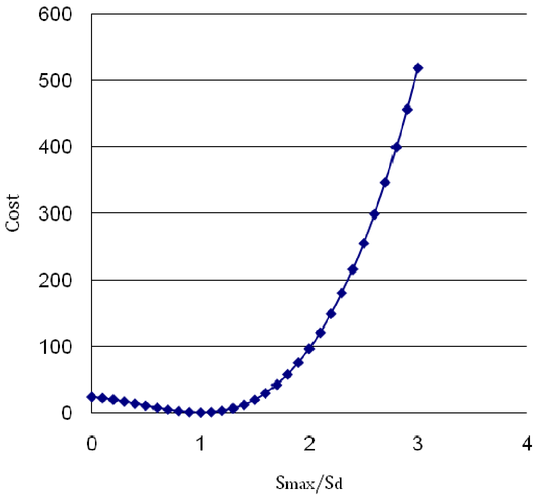


Figure 4.10 - Stress cost function behavior

The overall cost function can be calculated by assembling the stiffness and stress cost functions. A scaling constant (C) is used to normalize the values of the two cost functions. The scaling constant may also be used to increase or decrease the sensitivity of the cost function to the stress or stiffness cost values.

$$Cost = \left[\sum (K_i - K_d)^2 \right] + C \times \left[(\frac{S_{\max}}{S_d} - 1) \times \log(\frac{S_{\max}}{S_d} + 2) \right]^2 \tag{3.13}$$

4.3.3 Optimization

After defining the design parameters and cost function, an optimization method should be used to minimize the cost function. A general optimization algorithm will start with either predefined or randomly selected initial design variables. The cost is then calculated for the design variables, and the design variables are modified to form the next set of design variables. This algorithm is repeated until the cost is within a predefined acceptable range or the number of iterations reaches the maximum number of allowed iterations.

The major difference between various optimization methods is in the procedure used to modify the design parameters. From this point of view optimization methods can be divided to two major groups: gradient and non-gradient methods (Nocedal, et al.). Gradient methods use partial differentials and Hessian matrix to analyze the cost function's behavior and modify the design variables. These methods gradually converge to a local minimum point; however they require the cost function to be continuous and differentiable. On the other hand, non-gradient based methods such as genetic algorithm and simulated annealing use the gathered data from previous iterations and also random functions to modify the design parameters. Although the number of iterations for convergence might be more in non-gradient based methods but as they search a broader domain area, there is a higher chance of finding a global minimum. Moreover non-gradient based methods do not require continues cost functions.

In the spring optimization tool, since the maximum stress of the structure does not necessarily occur at the same location for all of the iterations; the cost function is not continues and a gradient based method cannot be used.

The optimization method that is used in this project is the Adaptive Simulated Annealing (ASA) method which is implemented in MATLAB by Stéphane Moins (Moins, 2002). The method is a nonlinear global method similar to Genetic Algorithm (GA), but in contrast with the GA method, each of the ASA iterations consists of only one generation of design variables which speeds up the optimization process. The simulated annealing method is derived from thermodynamics and is inspired by thermal annealing in solids. SA method was first introduced by Metropolis et al. in 1953. Metropolis' algorithm simulated the cooling of a solid material in a heat bath, i.e. annealing (Calculations by Fast Computing Machines Equation of State, 1953).

If a solid material is heated over melting point and then cooled down, the cooled down material's structure will depend on the rate of cooling. If the melted solid is cooled down gradually large solid crystals will be created however if the liquid is cooled down rapidly the formed crystal will contain imperfections. In Metropolis' algorithm the material is simulated with a system of particles that gradually loose temperature in the cooling process. The cooling process continues until the material converges to a steady frozen state.

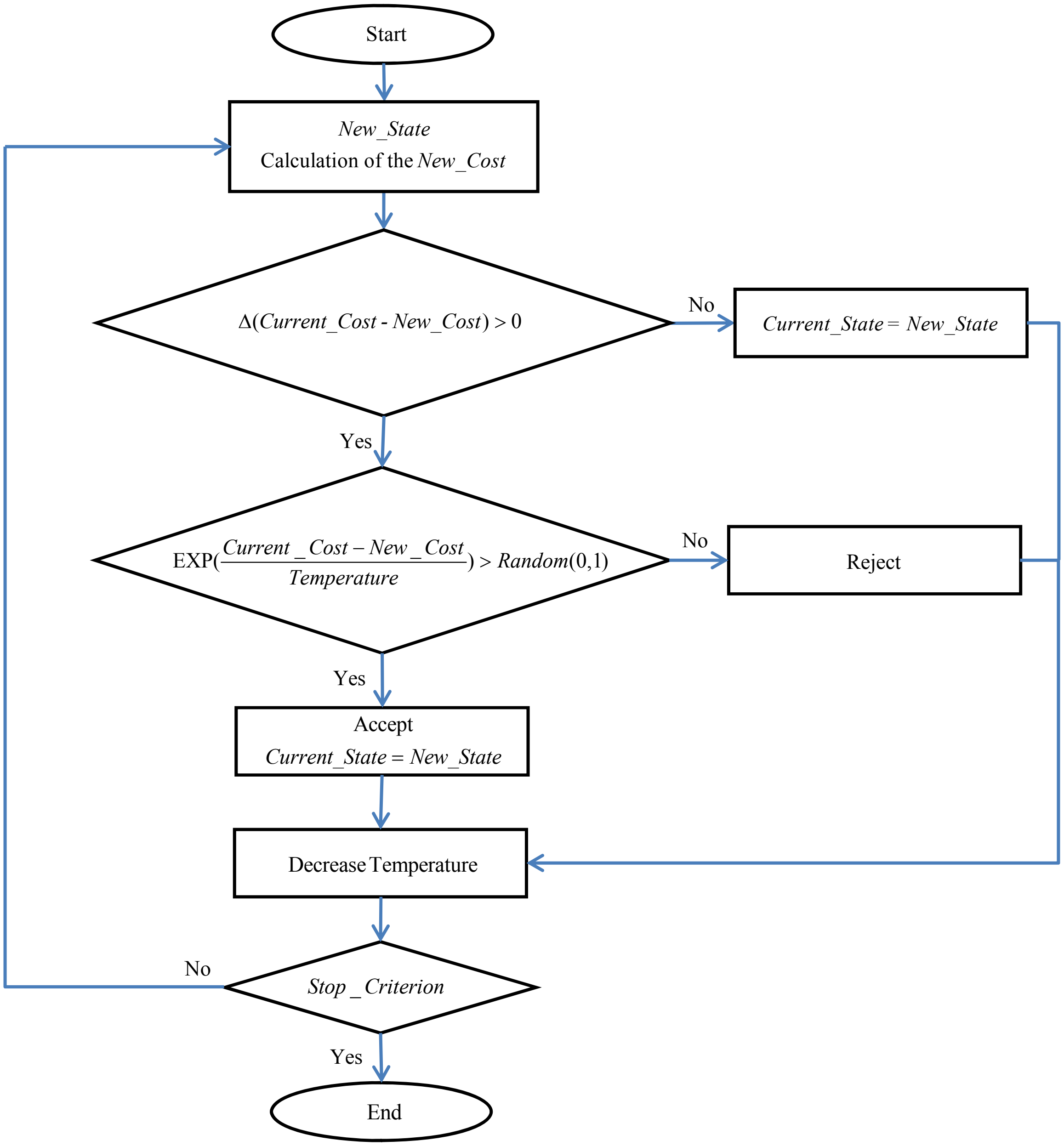
The optimization algorithm can be better visualized by a geographical terrain. Consider an optimization problem to find the deepest valley in a range. SA approaches this problem by assuming a bounce ball that can bounce from one valley to another over the mountains. The optimization starts at a high temperature which is identical to a high energy of the bounce ball and the ball can bounce over high mountains. However, as the optimization continues and the temperature drops, the bounce ball's energy decreases and therefore it cannot leap over high mountains anymore. Thus the ball will be trapped within a number of valleys. As the temperature further drops, the number of valleys decrease as well until one valley is selected as the deepest one (convergence) or the temperature drops below a specified limit (maximum number of iterations reached).

In the simulated annealing optimization method, the mountain range is described by the cost function. The algorithm then uses probability distributions and also the comparison of cost function values of the previous and current valleys to decide whether to stay in a new deeper valley or jump out of it. The temperature affects both the probability distribution and the cost function comparison (Van Laarhoven, et al., 1987).

Figure 4.11 shows the algorithm for the simulated annealing method. It starts with a manually inputted or randomly selected initial state. The algorithm continues by calculating the associated cost function for this state. This optimization is first started at a high hypothetical temperature. Then a new state is randomly generated. If the cost function for the new state is lower than that of initial state, the new state is accepted. This allows the algorithm to converge towards a local or possibly global minimum. However, in most multi-variable optimization programs, a number of local minimums exist which may trap the algorithm. To avoid the algorithm from getting stuck in such local minimums, a higher cost function value is accepted with a certain probability:

$$\mathrm{EXP}\left(\frac{Current\_Cost - New\_Cost}{Temperature}\right) > Random(0,1) \tag{3.14}$$

As it is demonstrated in equation (3.14) the acceptance probability of a state with higher cost function is dependent on the optimization temperature. Thus, there is a higher chance of acceptance at the beginning of the optimization where the temperature is higher.



Start
New_State
Calculation of the New_Cost
Δ(Current_Cost - New_Cost) > 0
No
Current_State = New_State
Yes
EXP(Current_Cost – New_Cost / Temperature) > Random(0,1)
No
Reject
Yes
Accept
Current_State = New_State
Decrease Temperature
No
Stop_Criterion
Yes
End

Figure 4.11 - Simulated annealing algorithm

For instance, in the iteration with the current state of *X* in Figure 4.12, the new state, *N1*, is always accepted; however there is a probability for the new set of *N2* to be accepted. This probability depends on the iteration's temperature and is higher for the beginning iterations. This probability decreases with the temperature as the optimization further advances.

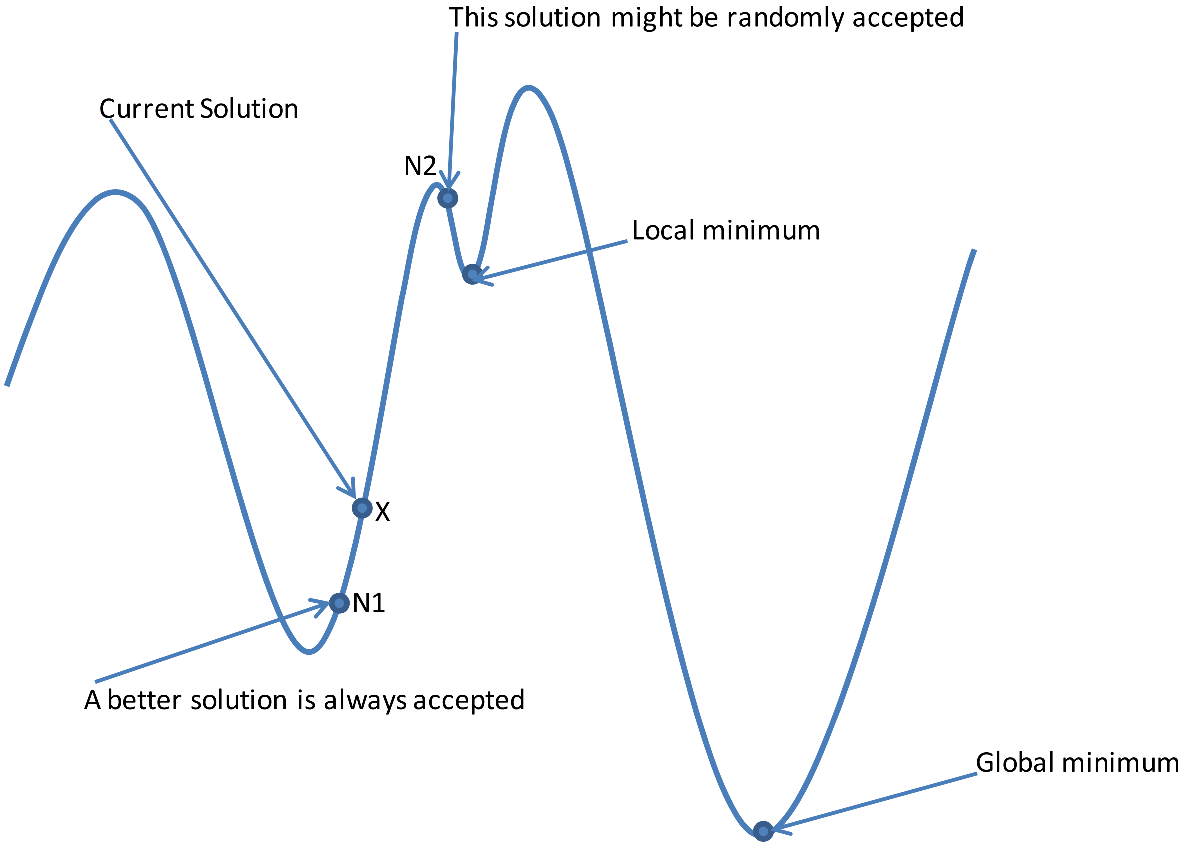
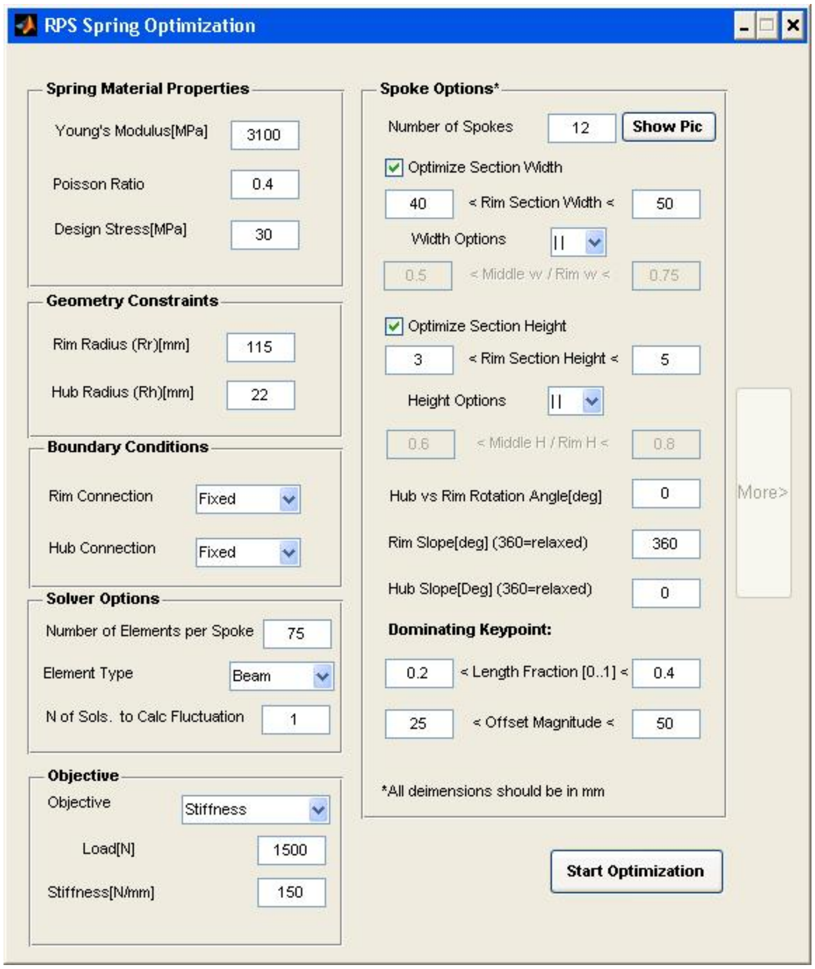

This solution might be randomly accepted
Current Solution
N2
Local minimum
X
N1
A better solution is always accepted
Global minimum

Figure 4.12 – Acceptance of new states

4.3.4 Computer Programming

The shape optimization is implemented in a program using ANSYS and MATLAB. A Graphical User Interface (GUI) is created in MATLAB to input the design variables and criteria (Figure 4.13). The GUI inputs material properties, geometrical constraints, and FEA and optimization options. After clicking the start optimization button, the program runs ANSYS in batch mode and sends the design information to ANSYS. The FEA model is created and solved in ANSYS, and the stiffness and stress information are sent back to MATLAB, and evaluated in the cost function. The following section describes the sequence of building and solving a spring model in ANSYS. A more detailed explanation on how to use the application and also an in-depth review of a sample ANSYS macro is presented in Appendix A.

The ANSYS code inputs the design variables from a text file created by MATLAB, then starts to build a single spring model. Depending on the selected element type in the solver options section, an ANSYS code for Beam, Shell, or Plane elements is run. The steps of creating a three point spline model using Plane elements is presented in this section: The macro begins with creating key points at locations defined by the design variables inputted from MATLAB (Figure 4.14). The next step in the ANSYS code is to create the spring's center line (Figure 4.14). The line is created using the defined key points and end slopes. If the end slopes are not defined, a relaxed spline - with second derivatives set to zero at end points - is created; otherwise the defined end slopes are applied to the spline. If the defined slopes are negative values, extra mid-points are automatically added to the model to prevent sharp bending at the end points of the spring.



Spring Material Properties

Young's Modulus[MPa] 3100

Poisson Ratio 0.4

Design Stress[MPa] 30

Geometry Constraints

Rim Radius (Rr)[mm] 115

Hub Radius (Rh)[mm] 22

Boundary Conditions

Rim Connection Fixed

Hub Connection Fixed

Solver Options

Number of Elements per Spoke 75

Element Type Beam

N of Sols. to Calc Fluctuation 1

Objective

Objective Stiffness

Load[N] 1500

Stiffness[N/mm] 150

Spoke Options*

Number of Spokes 12

☑ Optimize Section Width

40 < Rim Section Width < 50

Width Options ||

0.5 < Middle w / Rim w < 0.75

☑ Optimize Section Height

3 < Rim Section Height < 5

Height Options ||

0.6 < Middle H / Rim H < 0.8

Hub vs Rim Rotation Angle[deg] 0

Rim Slope[deg] (360=relaxed) 360

Hub Slope[Deg] (360=relaxed) 0

Dominating Keypoint:

0.2 < Length Fraction [0..1] < 0.4

25 < Offset Magnitude < 50

*All deimensions should be in mm

Figure 4.13 - Optimization GUI in MATLAB

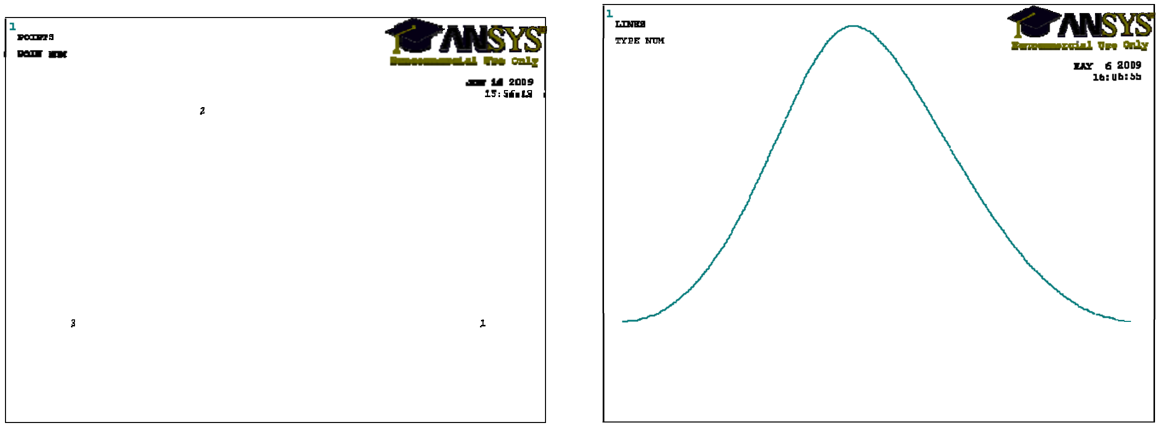


Figure 4.14 - Key point generation (left) Spring's centerline created with the defined key points and specified end slopes (right)

After creating the centerline which is basically the skeleton of the spring, the next step is to generate the spring's geometry by applying the specified section height and width through the length of the spring. For the sample three key point spring, the section information is inputted at five key locations and then linearly interpolated for the rest of the spring. The five locations include the three defined key points and also one key location between each two consequent key points.

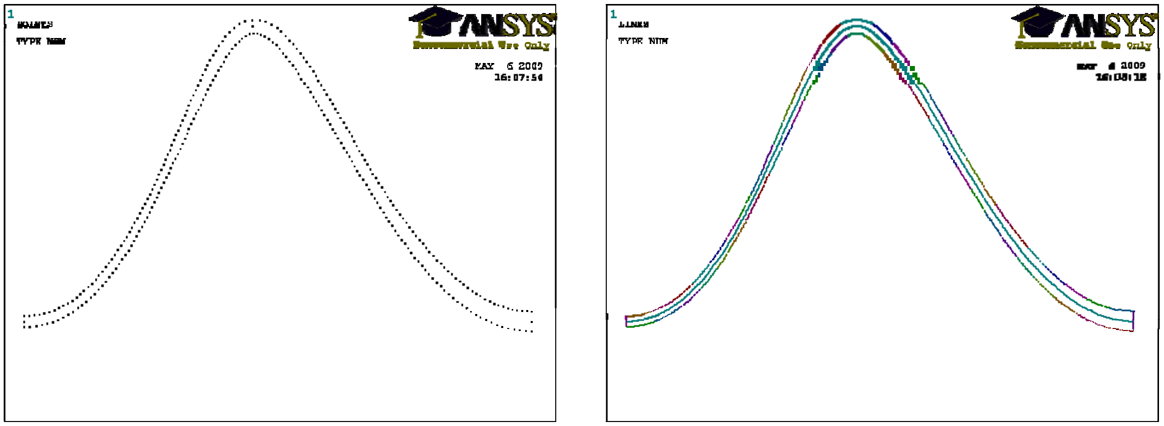


Figure 4.15 - Left: key points created along the centerline Right: spring's outer lines drawn through created key points

Having the section width and height values at each point along the centerline, a group of key points is created representing the spring's surface (Figure 4.15). Each six consequent key points are then selected to create the spring's outer lines (Figure 4.15). The key points are selected six by six as this is the maximum number of key points accepted by ANSYS' *BSPLIN* command (SAS IP, Inc., 2007).

The spring geometry is completed by creating an area surrounded by the drawn outer lines. The created area is shown in Figure 4.16. The FEA model continues by meshing the created geometry. The area is meshed with plane elements using the size and material information inputted from MATLAB (Figure 4.16). The elements are then patterned, using *EGEN* command to generate the rest of the springs as specified in the input file (Figure 4.17).

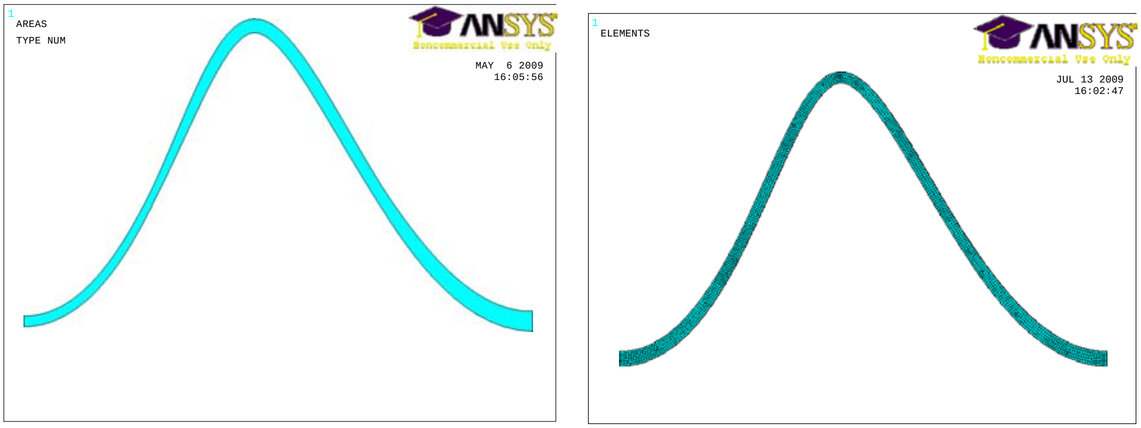


Figure 4.16 - Completed geometry for a single spring

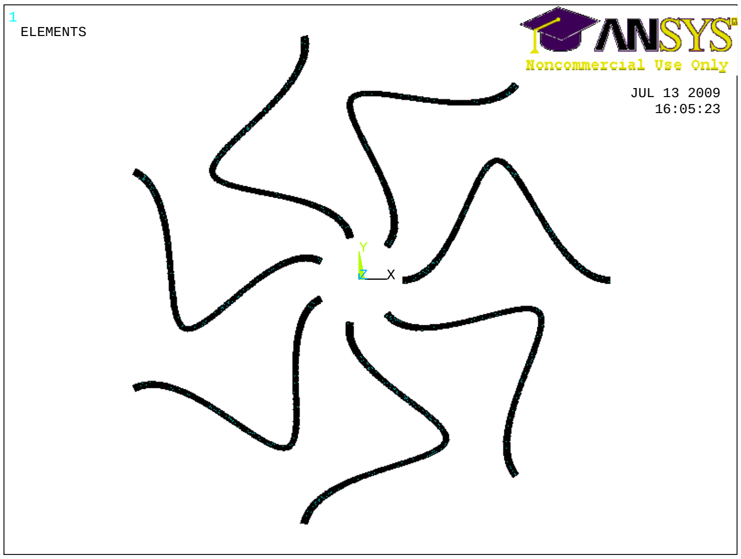


Figure 4.17 - The elements are patterned to generate the rest of the springs

After generating the elements, the boundary conditions should be applied. Depending on the inputs the springs may either have fixed connections or revolute joints at rim and hub. For the rim joint, in case of a fixed connection, all of the degrees of freedom of the nodes at the joint location will be set to a constant value of zero using ANSYS' *D* command (Figure 4.18). However for a revolute joint, as plane elements do not have rotational degrees of freedom, MPC184 elements should be used. MPC184 element is a Multi-Point Constraint element which can be used to model various joints or connections (SAS IP, Inc., 2007). To create a revolute joint, MPC184 elements are used as rigid binary links to connect the nodes located on the revolute joint's perimeter to a node created at the center of the revolute joint. The created node is then fixed to the ground. This combination allows the spring to rotate about the fixed node (Figure 4.18). For the hub connection, the process is similar however as the hub moves with respect to the rim - which is fixed to ground - the degrees of freedom of the nodes at the hub connection cannot be fixed. For a fixed connection between the springs and the hub, the nodes at the joint locations are coupled together using the *CP* command. This allows the nodes to move, however their relative positions would remain constant. For revolute joints at the hub, again the MPC184 element is used, but in contrast with the rim joints, the nodes at the revolute joints are not fixed to the ground, instead they are connected to a dummy node created at the center of the hub. This combination creates a rigid hub with revolute joints at spring connection points (Figure 4.19).

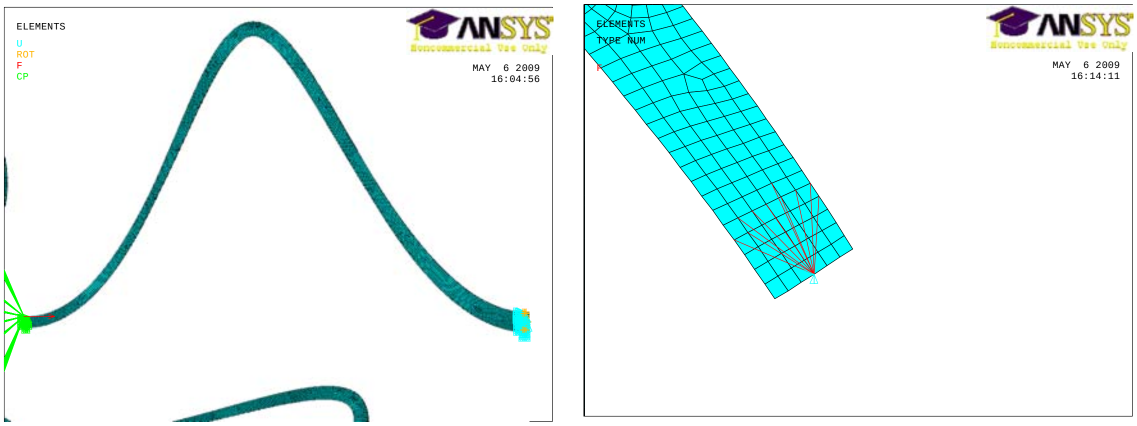


Figure 4.18 - Left: fixed connection Right: revolute joint

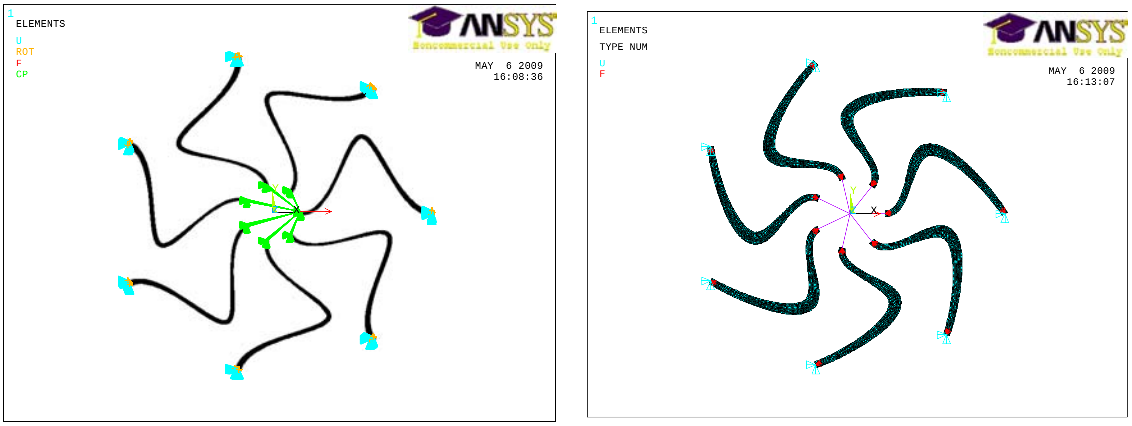


Figure 4.19 - Left: fixed connections at rim and hub Right: revolute joints at rim and hub

After creating the necessary boundary conditions, the load should be applied to the model. Depending on the input options, the load can either be a force or displacement at hub. If the springs are fixed to hub, the load is applied to the master node in the coupled node set. For revolute joint connections the load is applied to the dummy node created at the hub center. Furthermore, if the fluctuation minimizing option is selected in MATLAB, the space between two consequent springs is divided to two equal spaces. One of these spaces is selected and the load is applied in different rotation angles (Figure 4.20) and solved to speculate stiffness at different orientations of the wheel.

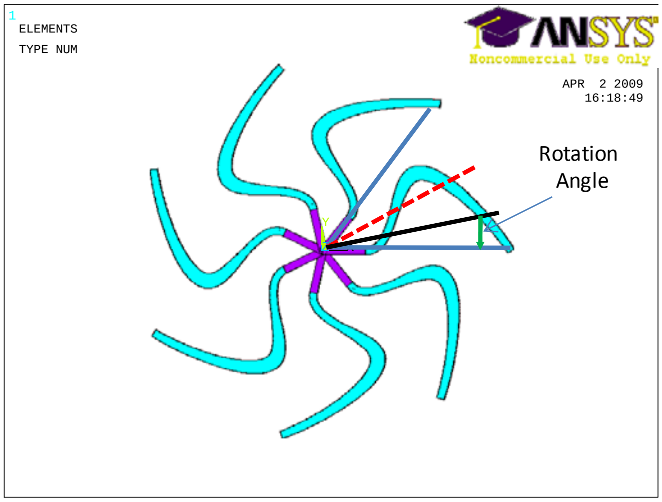


Figure 4.20 - If the fluctuation minimizing option is selected, the load is applied at different rotation angles in half of the space between two consequent springs.

Afterwards, the FEA model is solved in ANSYS. Figure 4.21 shows sample ANSYS results for fixed and revolute joint boundary conditions. The large deformation effects are taken into account setting the *NLGEOM,ON* command. Consequently the post processor module is used to calculate the structure's stiffness and maximum stress. The results are then written to a text file named "*ANS2MAT.dat*". The MATLAB code reads the ANSYS results from the text file and evaluates the cost function then adjusts the new design variables.

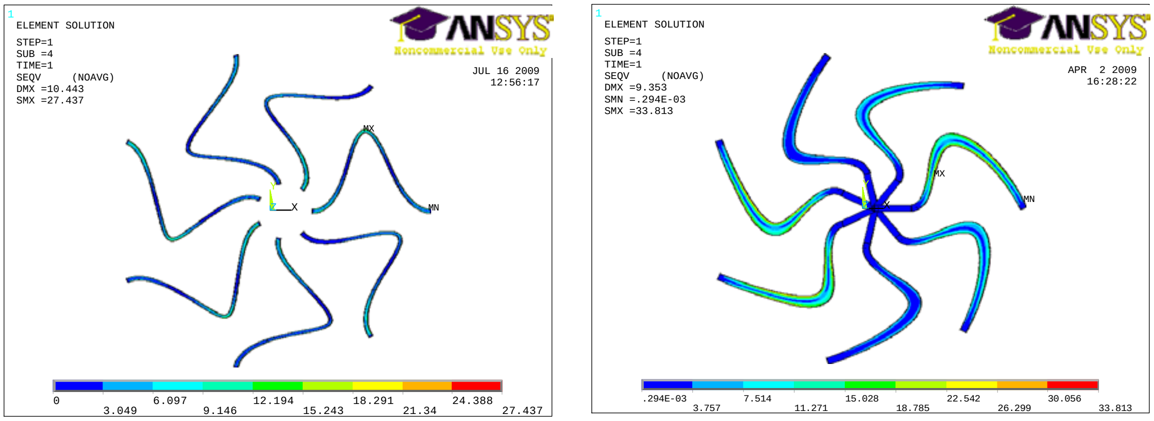


Figure 4.21 - Sample ANSYS results. Left: Fixed Right: Revolute

The optimization loop is then completed by sending the new design variables to ANSYS. This loop continues until the solution is converged to the best possible result or it is interrupted by the user.

4.4 Summary

In this chapter, energy and finite element analysis methods were used to optimize springs for rotating in-wheel suspension systems. The Castigliano's energy method is an analytical method which was used for optimization with MATLAB's gradient based optimization tools. Although the energy method is relatively fast but the springs are restricted to analytically defined shapes. In addition, the solution accuracy was also limited to small deformations. The spring optimization was then improved by using ANSYS as external finite element software. Using ANSYS in conjunction with an adaptive simulated annealing optimization code in MATLAB allowed for optimization of different spring shapes defined by a series of key points. Besides, large deformations and stress effects were also considered in the optimization. The final result was a spring optimization program with a graphical user interface in MATLAB for in-wheel suspension spring optimization.

Chapter 5
Prototyping and Testing

5.1 Introduction

This chapter includes two IWS designs developed using the optimization code developed in Chapter 4. The first design is a single part, fixed joint design, optimized for a proof of concept prototype and the second design is a practical flexible wheel with revolute joint springs designed for a manual hand-truck.

5.2 Fixed Connection Design

Fixed joint in-wheel suspension system is the simplest IWS design and is an ideal design for applications such as baby strollers, bicycles and gulf carts. The suspension is a single part which can be extruded with the wheel's hub and rim.

The following section presents the design and prototyping of a proof of concept fixed joint IWS optimized for a hand truck. An optimized wheel for a baby stroller is also presented at the end of the section.

5.2.1 Design Criteria

As a proof of concept design, the fixed joint prototype should demonstrate the practicality and simplicity of IWS. Thus, a soft, retrofit and easy to manufacture design is required. The fixed joint wheel prototype is intended to fit ordinary utility hand trucks; however, the load capacity is designed to be lower than the nominal 250lbs per wheel of hand trucks to emphasize the isolation capability and effectiveness of IWS. The overall dimensions and design targets of the fixed joint IWS are shown in Table 5.1.

.

Table 5.1 - Design criteria for fixed joint prototype

Parameter	Value
Nominal load	200N(45lbs) per wheel
Maximum load	400N(90lbs) per wheel
Stiffness	30N/mm
Wheel travel	15mm
Overall diameter	8"
Hub diameter	1.85"
Tread width	2"

5.2.2 Material Selection

DuPont Delrin 100 is selected as building material for the fixed joint prototype. Delrin is initially chosen because of its good machining properties and availability. However as it is discussed in section5.3.2, in addition to good machining capabilities, Delrin also has superior mechanical properties for a spring material. It has a metal like stress strain curve with a wide range of elastic behavior, high potential for storing elastic energy, high fatigue stress limit and also relatively stable mechanical properties at higher temperatures.

5.2.3 Optimization

At the time the fixed joint prototype was optimized, the optimization code was not as sophisticated as described in chapter 4. The code optimized the section height at only two locations: at the hub and the spring's midpoint; the spring's section height at rim was set equal to its section height at hub. Modeling with plane or shell element was not supported either; however as the section height for a fixed joint design is thinner than that of a spring with revolute joints, Beam elements can be used with sufficient accuracy and less simulation time. Table 5.2 shows the design variables and optimization parameters used for optimizing the wheel. The optimization results are displayed in Table 5.3.

Table 5.2 - Design parameters for fixed joint wheel

Parameter	Value
Hub radius (R_H)	22mm
Rim radius (R_R)	90mm
Spring width (B)	40mm
Number of spokes (N_SPK)	7
Spline end slop at hub (SLP_H)	90^{o}
Spline end slope at rim (SLP_R)	90^{o}
Hub vs. rim rotation angle (PHI)	0^{o}
Hub joint type	Fixed
Rim joint type	Fixed
Nominal force (FRC)	200N (250lbs)
Design stress at nominal load (S_D)	30MPa
Target Stiffness (K_D)	30N/mm
Material Young's modulus (MAT_E)	3100MPa
Material Poisson ratio (MAT_V)	0.35
Element type	Beam
Initial number of elements on spring's centerline (N_EL)	240
Section height at hub (H_HUB)	To be optimized
Section height at rim (H_RIM)	Equal to H_HUB
Section height at bending point (H_MID)	To be optimized
Offset magnitude (M_V)	To be optimized
Length Fraction (L_R)	To be optimized

Table 5.3 - Optimization results

Parameter	Value
Stiffness	27N/mm
Max stress at nominal load	28MPa
Section height at hub (H_HUB)	2.8mm
Section height at rim (H_RIM)	2.8mm
Section height at bending point (H_MID)	1.8mm
Offset magnitude (M_V)	53mm
Length Fraction (L_R)	0.3

5.2.4 Design finalization, CAD, Prototyping, and Testing

Further details such as, detailed hub and rim designs, bearing housings and rubber tread are required to convert the optimized spring solution to a practical wheel. Figure 5.1 shows a solid model prepared in SolidWorks. The prototype is made of a solid Delrin disc which is first machined to create the hub and rim details. Then the extra material is removed by water-cutting. As it seen in Figure 5.1, three rubber O-rings are fitted on the rim to perform as the rubber tread. Although the isolation and grip of the O-rings are less than those of a rubber tread but this design avoids the complicacy of molding rubber over the Delrin wheel in the proof of concept prototype.

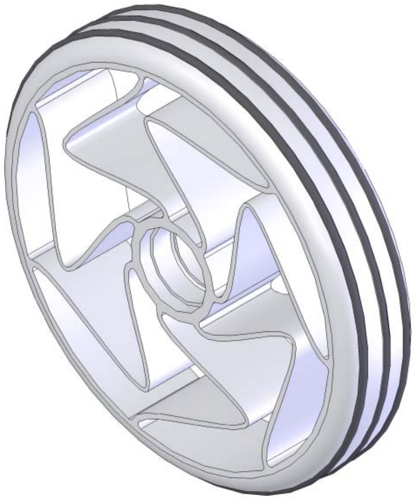


Figure 5.1 - CAD model for fixed joint prototype

Figure 5.2 displays a manufactured fixed joint prototype. Prototypes are also sent to the project's industrial partner for experimental tests on hand trucks and evaluation of mechanical properties. Figure 5.3 displays the designed prototype installed on a hand truck. The hand truck is loaded with 130lbs and driven over and dropped off street curb. Figure 5.3 shows the instant that the loaded hand truck is dropped off a curb. The photo demonstrates the performance of the IWS and the shock absorbance and flexibility of the structure is apparent.

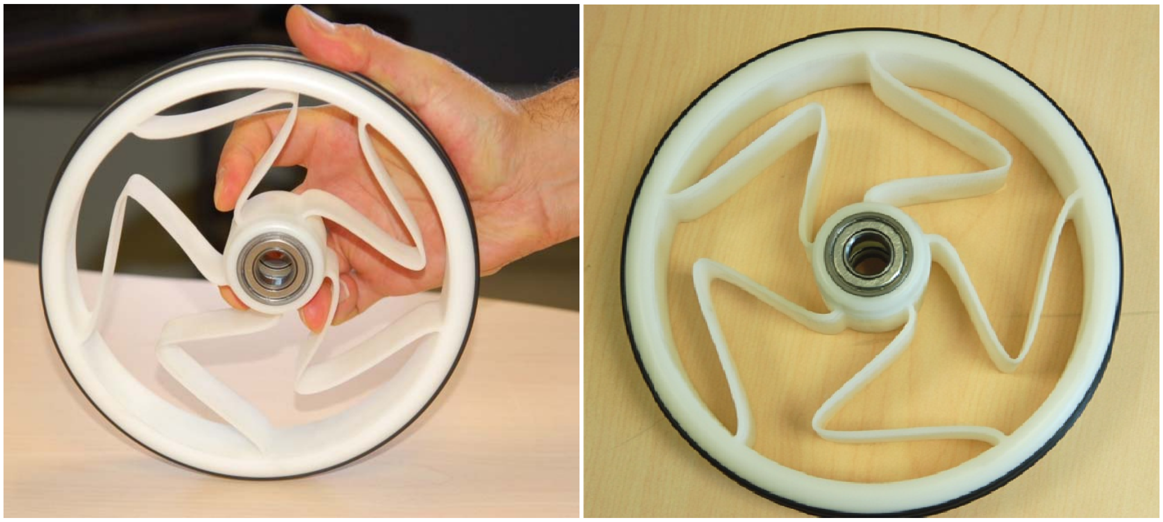


Figure 5.2 - Optimized flexible wheel with fixed joints

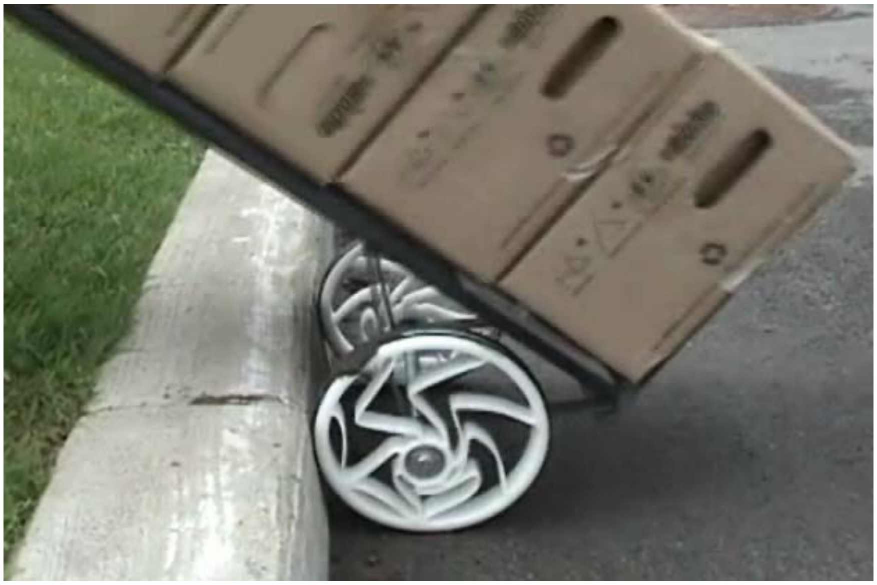


Figure 5.3 - The prototype installed on a hand truck

In addition to the hand truck test, the prototype is also investigated for mechanical properties at Cooper-Standard Co. Figure 5.4 shows the test setup and test results from Cooper-Standard Co. As it is seen from the graph, the overall stiffness of the wheel is 27.52N/mm which shows less than 2% of error between the optimization and experimental results.

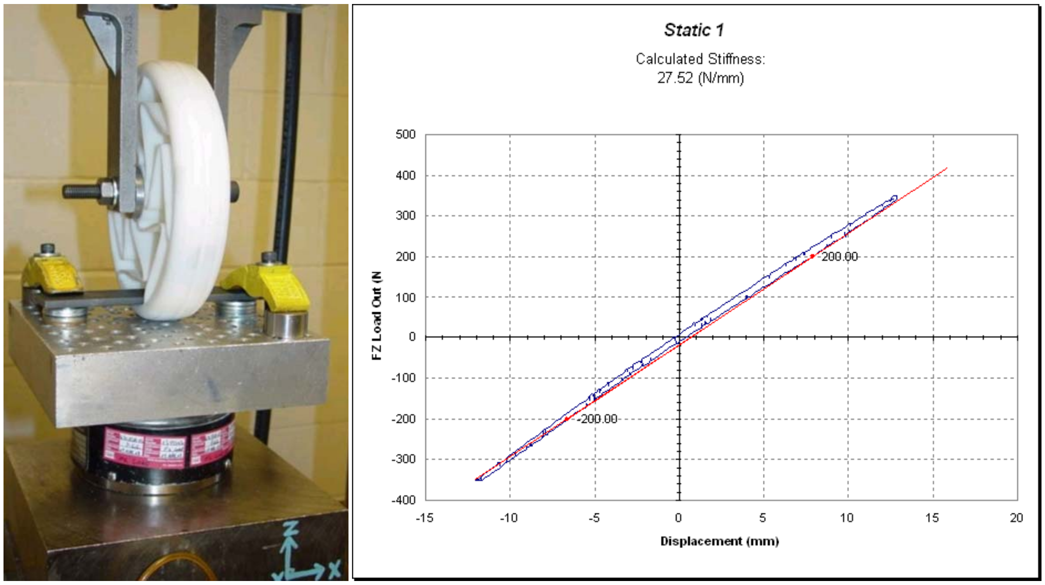


Figure 5.4 - Test setup and results from Cooper-Standard Co.

5.2.5 Fixed Joint Design for Baby Stroller

Low working speed, low load and also relatively large wheels make baby strollers a good candidate for IWS application. Figure 5.5 shows a CAD model designed for a baby stroller. The studied baby has two 16" rear wheels and a single 12.5" front wheel. The nominal working load including the weight of the stroller is about 40Kg and the maximum load is 100Kg. The designed IWS has similar spring units in both rear and front wheels. The suspension system allows for 25mm of wheel travel from each side. The material used for optimization is Tecast Vekton® 6XAU nylon which is less expensive than Delrin and has the ability to be over molded with other polymers despite Delrin.

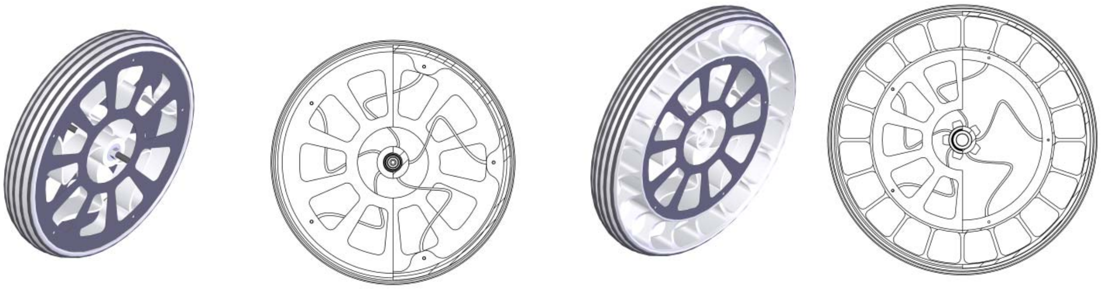


Figure 5.5 - Fixed joint IWS design for baby stroller; Left: front wheel Right: rear wheel

5.3 Revolute Joint Design

In addition to fixed joints, revolute joints can be used in an in-wheel suspension system to connect the springs to the wheel's hub and rim. Although a fixed joint design has advantages such as fewer parts, lower weight and ease of fabrication, but the stress concentration at fixed joints limits the load capacity. Thus, for applications requiring higher load capacities, revolute joints can be used to decrease the stress levels. The developed optimization code allows the user to select fixed or revolute joints or a combination of the two joint types as the applied boundary condition.

Design, optimization, prototyping and testing procedure of an in-wheel suspension system with revolute joints is described in the following sections. The goal is to design a flexible wheel for a utility hand truck addressing all the requirements defined by the hand truck manufacturer.

5.3.1 Design Criteria

Depending on the working environment hand truck manufacturers supply various types of wheels that can be installed on hand trucks. Solid wheels can be used for paved surfaces and higher load capacities while pneumatic wheels are available for rough surfaces at the cost of a higher rolling resistant and the need to maintain the wheel at the specified inflation pressure. There are also multipurpose plastic or metal wheels with thick rubber treads providing less shock absorption and rolling resistance in comparison to pneumatic tires (Figure 5.6). IWS can be implemented in hand truck wheels to fill the gap between these products. The target is to design a wheel with isolation benefits of pneumatic tires, and load capacity and lower maintenance requirements of solid wheels. The rolling resistance is improved by elimination of the pneumatic tire or a thick rubber tread.

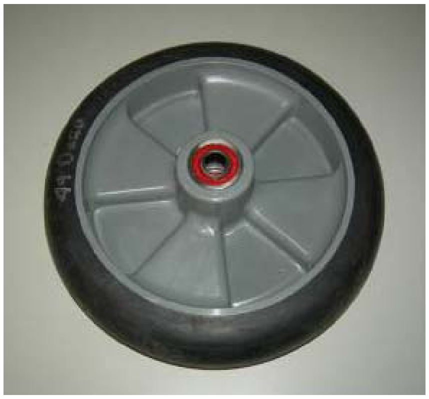


Figure 5.6 - Magline 1030 hand-truck wheel

The designed wheel must fit current hand trucks without any modification of the connecting mounts. Table 5.4 shows the general load and geometry properties that should be met by the design. In addition to these properties, the designed wheel has to also fulfill the static and dynamics requirements specified in Table 5.5.

Table 5.4 - Design parameters for hand truck wheel

Parameter	**Value**
Nominal load	250lbs per wheel
Overall diameter	10"
Hub diameter	2"
Tread width	2"

Table 5.5 - Static and dynamics specifications required by Magline

Test	**Requirement**
Rolling Resistance	Required force to start/maintain motion
Coefficient of Friction Test	Required force to pull a hand-truck loaded with 1200lb while wheels are locked
Static Load	1200 lbs
Dynamic Load Test	50,000 cycle at 300/400/500lbs
Side Load Test	Apply a lateral 300lb load to the wheel and measure the deflection
Vertical Impact Test	Minimum 300lbs*2” = 34J
45 deg Oblique Impact Test	Energy absorption: 69J
Abrasion Resistance Test	Evaluate the caster’s resistance to rolling over an abrasive surface
Peel off Test	Determine the load at which the rubber separates from the caster
Coefficient Of Restitution	Drop a 2.3lb ball from 72” on the caster and measure the bouncing height

5.3.2 Material Selection

Before starting the spring optimization the spring material should be selected. Various properties such as working condition, environment, temperature, load, deflection, fatigue life, and testing methods should be known to select the right spring material (Yamada, et al., 2007). A general good spring material should have a high capacity to store energy. The elastic energy that a spring material can store by its volume unit (A preliminary approach to composite beam design using FEM analysis, 1990) is given by equation (3.15).

$$S = \frac{1}{2}\frac{\sigma^2}{E} \tag{3.15}$$

Where

σ = Max stress

E = Young's modulus

A spring's capacity to store elastic energy is affected by amount of material used in its manufacturing too. The specific strain energy (Analysis and optimization of a composite leaf spring, 2003), S_s, can be used to calculate the elastic energy stored in a spring material's unit mass.

$$S_s = \frac{1}{2}\frac{\sigma^2}{\rho E} \tag{3.16}$$

Where

ρ = Material density

In general a material with a high yield stress to modulus of elasticity ratio is considered a good spring material. Furthermore the material should have relatively constant mechanical properties in the working environment such as temperature range. For springs used in cyclic applications the material's fatigue limits has to be considered as well.

In addition to the specific strain energy, a spring factor, *SF*, is defined to compare different materials for IWS application. *SF* is defined by investigating an elastic beam in pure bending condition. The parameter is defined by comparing two different materials to build a spring with a constant stiffness and section width. Each material has its own Young's modulus and maximum stress limit and the beam's section height is modified accordingly to achieve the same stiffness. It can be shown that an elastic beam's stiffness is proportional to the material's Young's modulus, the beam's section width and the cubic power of the section height:

$$k \propto Ebh^3 \tag{3.17}$$

Where

k = Stiffness

b = Section width

h = Section height

Furthermore, the stress level in the beam is proportional to the inverse of its section height. Thus, the materials' stress levels can be related to the section heights as shown in equation (3.18).

$$\frac{\sigma_1}{\sigma_2} \propto \left(\frac{h_2}{h_1}\right)^2 \tag{3.18}$$

Combining equations (3.17) and (3.18) the spring factor for each material can be defined by equation (3.19). Comparing *SF* values for two springs with the same stiffness and width and different materials; a higher *SF* value indicates a lower stress to max stress ratio.

$$\begin{cases} h_2 = \sqrt[3]{\dfrac{E1}{E2}}h_1 \\ \sigma_2 = \left(\sqrt[3]{\dfrac{E2}{E1}}\right)^2 \sigma_1 \end{cases} \Rightarrow SF = \frac{\sigma_{\max}}{\left(\sqrt[3]{E}\right)^2} \tag{3.19}$$

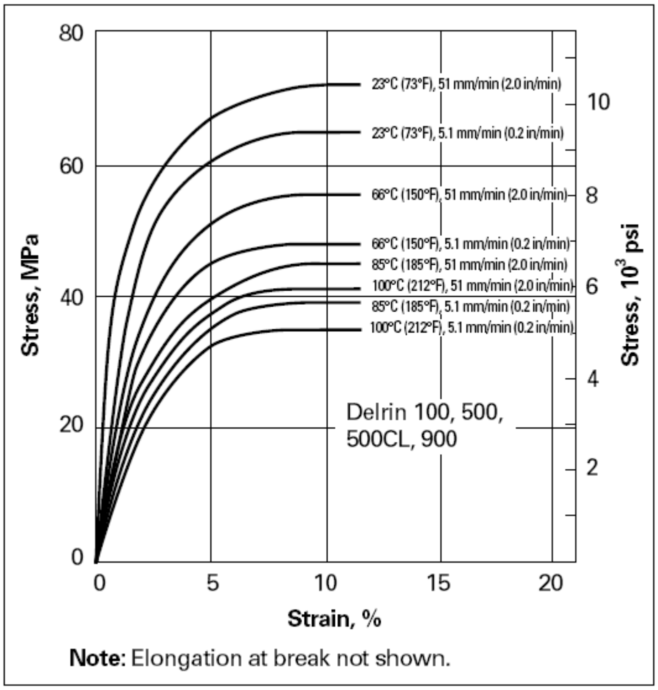
Spring factor values for various materials are shown in Table 5.6. The first column indicates the material name and information, the second column shows *SF* values when the maximum design stress is set to the material's yield stress and finally the third column displays *SF* values when the maximum design stress is limited to the material's fatigue stress limit.

Table 5.6 - Spring factor for different materials (DuPont Corporatrion, 2008) (BASF Corporation, 2007)

Material	**Spring Factor (*SF*$_y$) $\sigma_{max} = \sigma_y$**	**Spring Factor (*SF*$_{ssy}$) $\sigma_{max} = \sigma_{ssy}$**
Spring steel	0.4246	0.145
BASF A3WG7	0.4206	0.160
BASF B3ZG6	0.3541	0.150
BASF A3WG10	0.3341	0.150
DuPont Delrin 100	0.3057	0.151
BASF N2200G53	0.2875	0.140

As it is shown Table 5.6, spring steel has the highest SF_y value among the selected materials however steel strips are only supplied with constant section heights and therefore a height optimization cannot be applied to a steel spring. Furthermore, for cyclic applications, spring steel's SF_{ssy} is only higher than that of BASF's Ultraform N2200G53. The BASF Ultramids (A3X and B3X) have the highest SF_y values after spring steel. Although the Ultramids have good *SF* values for both yield and fatigue limits but their high capacity to absorb humidity (more than 2%) changes their mechanical properties drastically (BASF Corporation, 2007). Among the materials shown in Table 5.6 DuPont's Delrin and BASF's Ultraform N2200G53 both have relatively stable mechanical properties both in dry and conditioned situations. Thus, Delrin which has a higher *SF* value is preferred over the N2200G53 Ultraform.

In addition to good mechanical properties as a spring material, Delrin has also an acceptable behavior at higher temperatures. Figure 5.7 displays the stress strain curve for Delrin at different temperatures. As it is seen from the curves Delrin maintains about 70% of its mechanical strength even at temperatures as high as 66^{o}C.



Note: Elongation at break not shown.

Figure 5.7 - Delrin behavior at various temperatures (DuPont Corporatrion, 2008)

Delrin's mechanical properties are verified by a tensile test before using it in the optimization program. Figure 5.8 shows the tensile sample and the test results. As it is shown in the stress-strain graph, Delrin's tensile behavior is close to that of ductile metals such as Aluminum. The ultimate stress is about 73MPa and the material shows elastic behavior for stress values less than 55MPa.

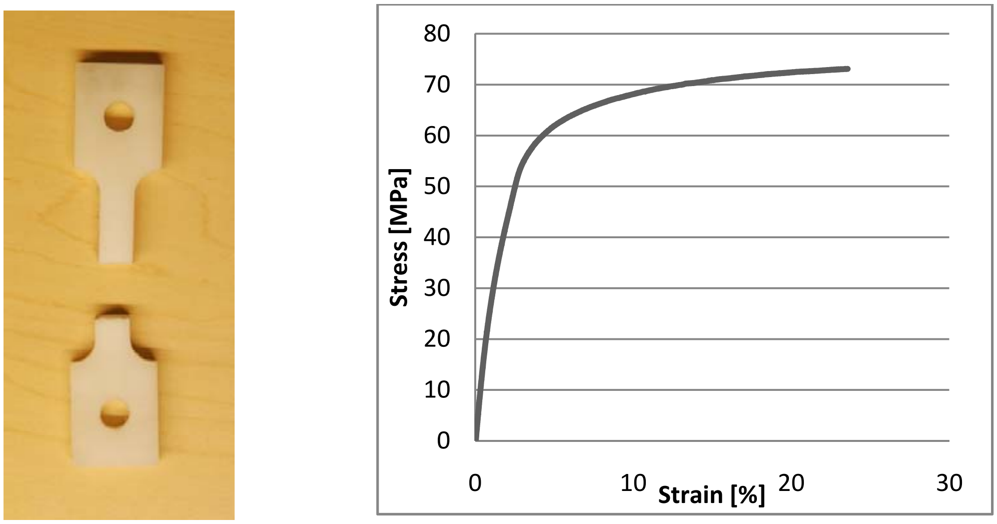


Figure 5.8 - Tensile test results for Delrin

5.3.3 Optimization

After defining the design criteria and selecting the spring material, the design parameters can be specified for the optimization program. Table 5.7 shows the design parameters that are used for the Magline hand truck wheel optimization. The design parameters are also graphically presented in Figure 5.9.

Table 5.7 - Design parameters for hand truck wheel

Parameter	Value
Hub radius (R_H)	26mm
Rim radius (R_R)	115mm
Spring width (B)	50mm
Number of spokes (N_SPK)	7
Spline end slop at hub (SLP_H)	0^o
Spline end slope at rim (SLP_R)	360^o (relaxed)
Hub vs. rim rotation angle (PHI)	0^o
Hub joint type	Revolute
Rim joint type	Revolute
Revolute joint radius (R_J)	5mm
Nominal force (FRC)	1113N (250lbs)
Design stress at nominal load (S_D)	30MPa
Target Stiffness (K_D)	135N/mm
Material Young's modulus (MAT_E)	3100MPa
Material Poisson ratio (MAT_V)	0.35
Element type	Plane
Initial number of elements on spring's centerline (N_EL)	150
Section height at hub (H_HUB)	To be optimized
Section height at rim (H_RIM)	To be optimized
Section height at bending point (H_MID)	To be optimized
Section height at the point between hub and bending point (H_H_M)	To be optimized
Section height at the point between rim and bending point (H_H_M)	To be optimized
Offset magnitude (M_V)	To be optimized
Length Fraction (L_R)	To be optimized

The optimization program is run for different number of springs. Higher number of springs results in less stiffness fluctuations, but more fabrication and assembly cost and complication. Furthermore, higher number of springs decreases the space available for each spring which results in a stiffer structure. The best optimization results for the hand truck wheel are achieved by using seven springs. Table 5.8 shows the results for the hand truck springs after more than 5,000 iterations. The resulting stiffness is 129 N/mm (target: 135 N/mm) and the maximum stress is 31MPa (target: 30MPa).

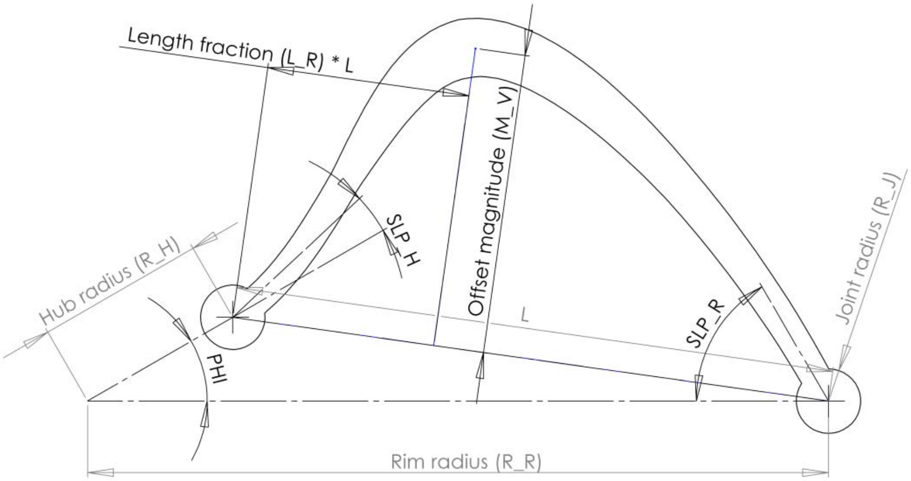


Figure 5.9 - Design parameters for the hand truck wheel

Table 5.8 - Optimization results

Parameter	Value
Section height at hub (H_HUB)	4.5933mm
Section height at rim (H_RIM)	4.6017mm
Section height at bending point (H_MID)	9.2025mm
Section height at the point between hub and bending point (H_H_M)	4.8210mm
Section height at the point between rim and bending point (H_R_M)	6.0606mm
Offset magnitude (M_V)	46.1312mm
Length Fraction (L_R)	0.31625

5.3.4 Design finalization, CAD, FEA Modeling, Prototyping, and Testing

The springs optimized in section 5.3.3 are implemented in a practical design to fit the hand truck without any modification. The design includes rigid aluminum rim and hub components and stainless steel side plates. The side plates are bolt to the rim to hold the Delrin springs in place. Furthermore, the maximum wheel travel in unexpected load conditions is restricted to design limits by a stainless steel stopper plate attached to the rim (Figure 5.10). Moreover, the outer surface of the rim is covered with rubber for a better grip and isolation. Detailed drawings for the designed prototype are given in Appendix B.

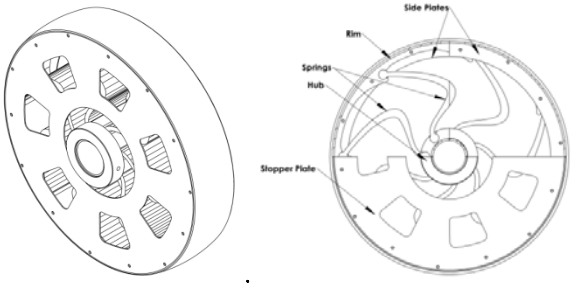


Figure 5.10 - CAD model for revolute joint hand truck prototype

All of the components are also modeled in ANSYS to ensure Magline test requirements are met. The tests include static vertical and lateral maximum load, and also dynamics vertical and oblique drop tests. Figure 5.11 shows the FEA models for some of the components.

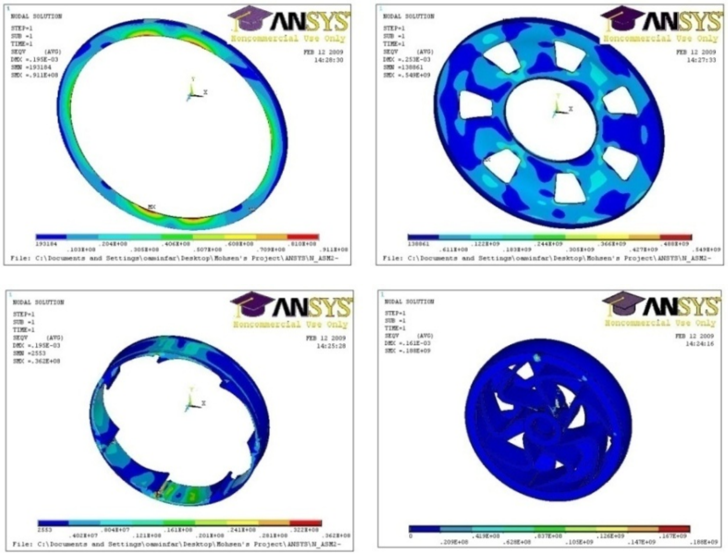


Figure 5.11 - FEA results

The prototyping procedure is followed by machining a single spring out of Delrin and testing it for static stiffness (Figure 5.12). The spring is also tested for its behavior in different temperatures. The results show less than 10% of stiffness change at 40°C and about 25% at 60°C. Furthermore a fatigue test is done to check the durability of the spring. Figure 5.13 shows two cycles of the fatigue test at the 1st and 5000th cycles. The graph demonstrates the constant behavior of the spring. The fatigue test was continued for 2.5 million cycles and the spring lasted without any visible signs of mechanical failure at the nominal 9mm of wheel travel.

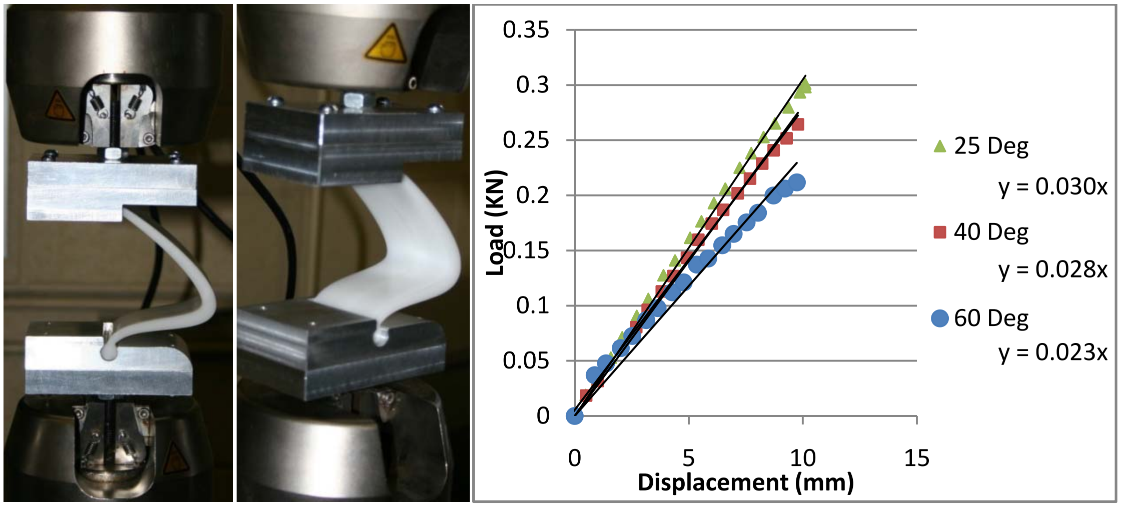


Figure 5.12 - Test setup and results for a single Delrin spring

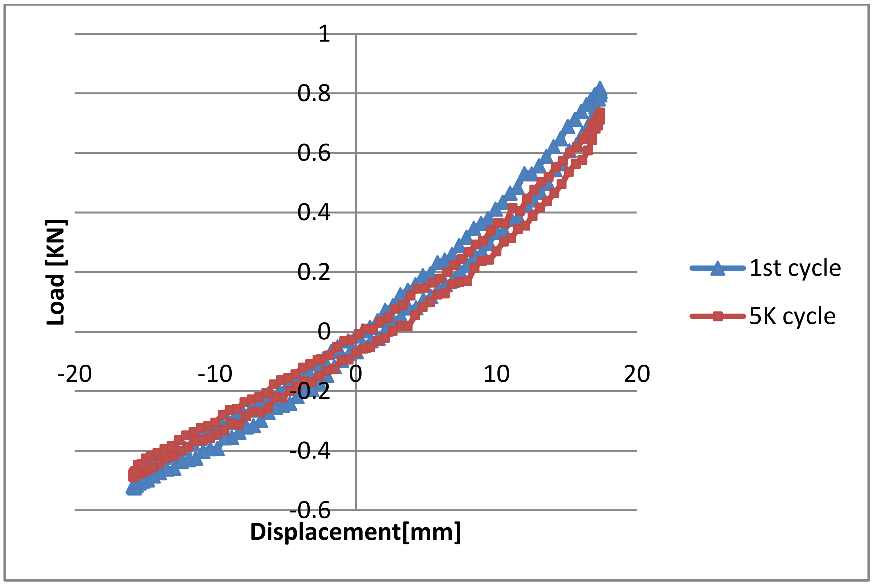


Figure 5.13 - Fatigue test

After verifying the optimization results with a single spring prototype, a whole wheel prototype is built. The manufactured prototype is shown in Figure 5.14. As it is seen from the pictures, a groove is also added to the wheel's hub allowing the insertion of a cotter pin. This cotter pin required by the hand truck manufacturer for fixing the wheel on the hand truck's shaft.

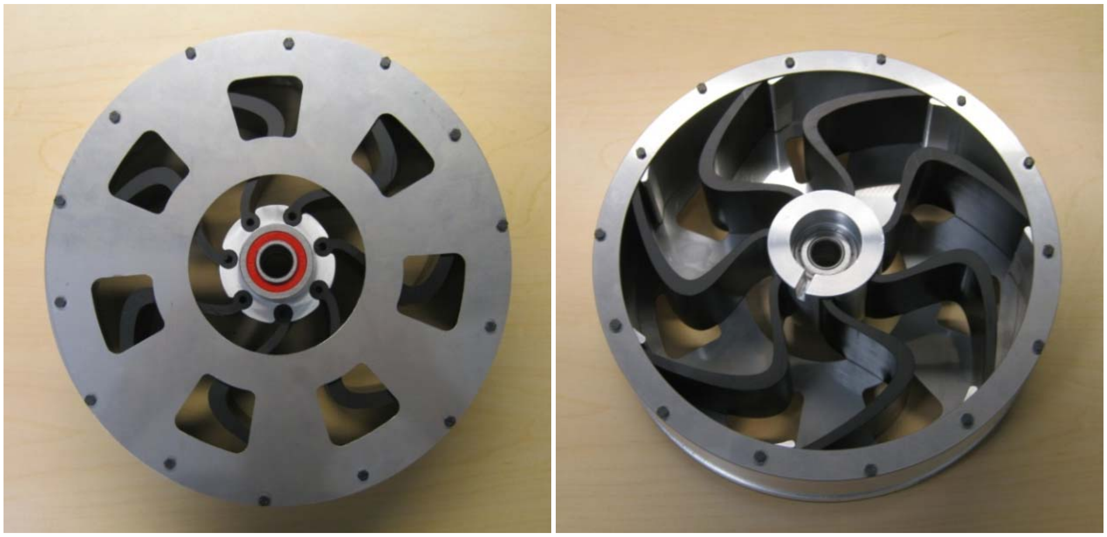


Figure 5.14 - Built hand truck wheel prototype

The built prototype is tested at the University of Waterloo’s high pressure laboratory. The test setup and results are shown in Figure 5.15. Figure 5.16 shows the test results for static and dynamics loadings. The static stiffness of the prototype is measured to be 136N/mm (target: 135 N/mm, optimization result: 129N/mm). The prototype is also tested for the dynamics loading requirements specified by Magline. The wheel is tested for 50,000 cycles at 300, 400 and 500lbs loads. The prototype has passed the tests without any signs of mechanical failure to the springs or rigid components. Figure 5.16 also shows one of the cycles of the dynamics loading tests. The hysteresis effect visible in the graph is due to the friction at the revolute joints and also the structural damping effects of the flexible components.

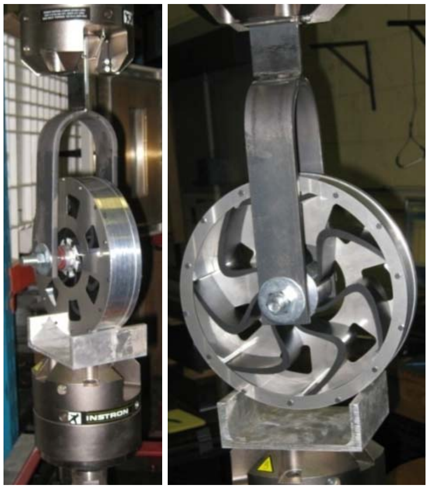


Figure 5.15 - Test setup for the hand truck wheel

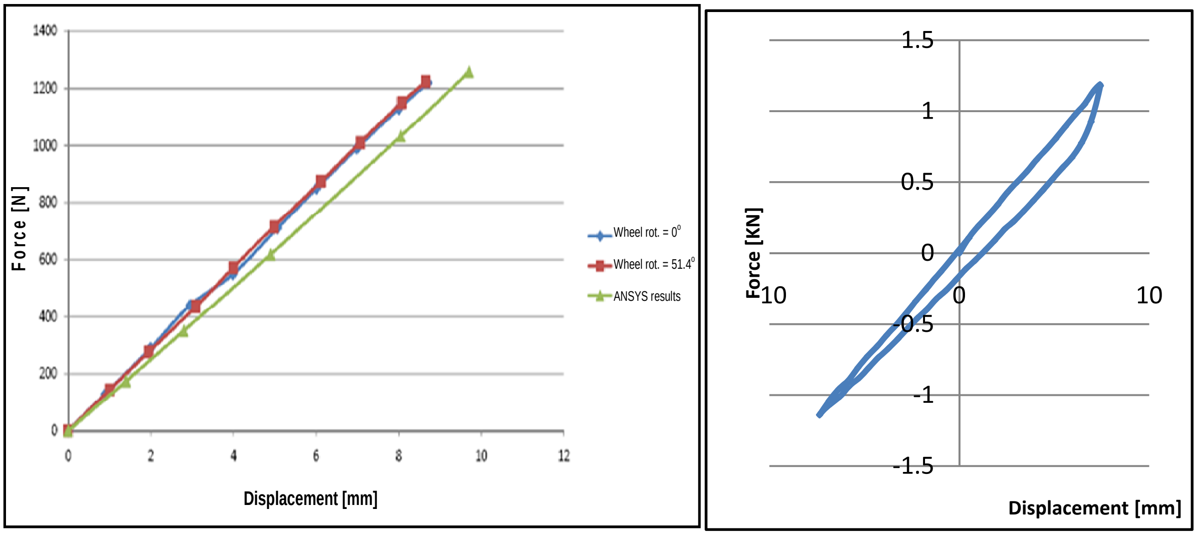


Figure 5.16 - Static (left) and dynamics (right) test results

5.4 Summary

In this chapter, the spring optimization tool developed in Chapter 4, was used to design and optimize two prototypes for the IWS. The prototypes were then finalized and built at the University of Waterloo's machine shop. The built prototypes were then tested at the University of Waterloo's high pressure lab and also at the project's industrial partner's facilities. The experimental results matched the simulation results with less than five percent of error and verified the program that was developed for spring optimization.

Chapter 6
Conclusions and Future Works

6.1 Conclusions

In this thesis a novel suspension system was studied. The studied suspension system is placed inside a vehicle's wheel and is called an in-wheel Suspension System (IWS). The project focused on investigation, optimization, and design of a prototype for a rotating IWS that can be used for low speed and low load vehicles such as hand trucks, wheelchairs, and bicycles.

First, a mechanism based IWS which was previously designed for a wheelchair by static methods was investigated for dynamics response in Chapter 3. The dynamics simulations demonstrated the influence of different design parameters on the IWS. The results from Chapter 3 showed the feasibility of designing a rotating suspension with a linear stiffness rate and minimal fluctuations. The results also concluded that a rotating suspension system's performance is undermined at higher speeds. Moreover, the effectiveness could be improved by reducing the mass of the suspension's rigid links.

Afterwards, replacing the rigid suspension components with flexible structures was studied. This was done using energy and finite element methods in Chapter 4. The studies in chapter 4 conducted to the development of a spring optimization program. The developed interface can be used to optimize general two dimensional springs for an IWS through a graphical user interface in MATLAB. The spring optimization tool developed in Chapter 4 was then verified by optimizing, building, and testing of two prototypes in Chapter 5. Experimental results verified the simulation results with less than five percent of error.

6.2 Future Works

The developed IWS shows the feasibility and practicality of the IWS concept. However, employing the current design in a commercial application requires further optimization and tuning. Moreover, this research has studied the simple case of a non-powered, rotating IWS for low speed vehicles; however future researches can be conducted on more complicated IWS designs such as powered applications and designs for road vehicles.

The main challenge in the powered rotating in-wheel suspension design is achieving infinite rotational stiffness in addition to the requirements of a non-powered IWS. In a rigid mechanism design, the rotational stiffness can be attained by using parallel mechanisms. However, maximizing the rotational stiffness for a compliance design is a more significant challenge. Figure 6.1 shows a primary powered design inspired by the parallel mechanism design. Preliminary results show enhancements in the rotational stiffness, yet the unsymmetrical configuration of the spokes about the wheel axis yields different rotational stiffness rates in clockwise and counterclockwise directions.

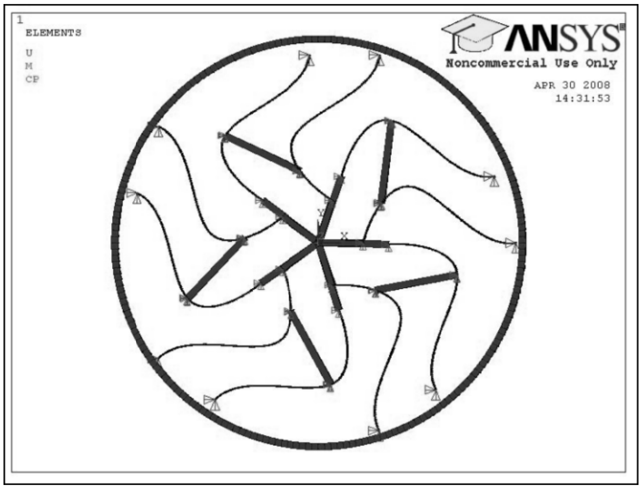


Figure 6.1 - Primary powered IWS

Appendix A

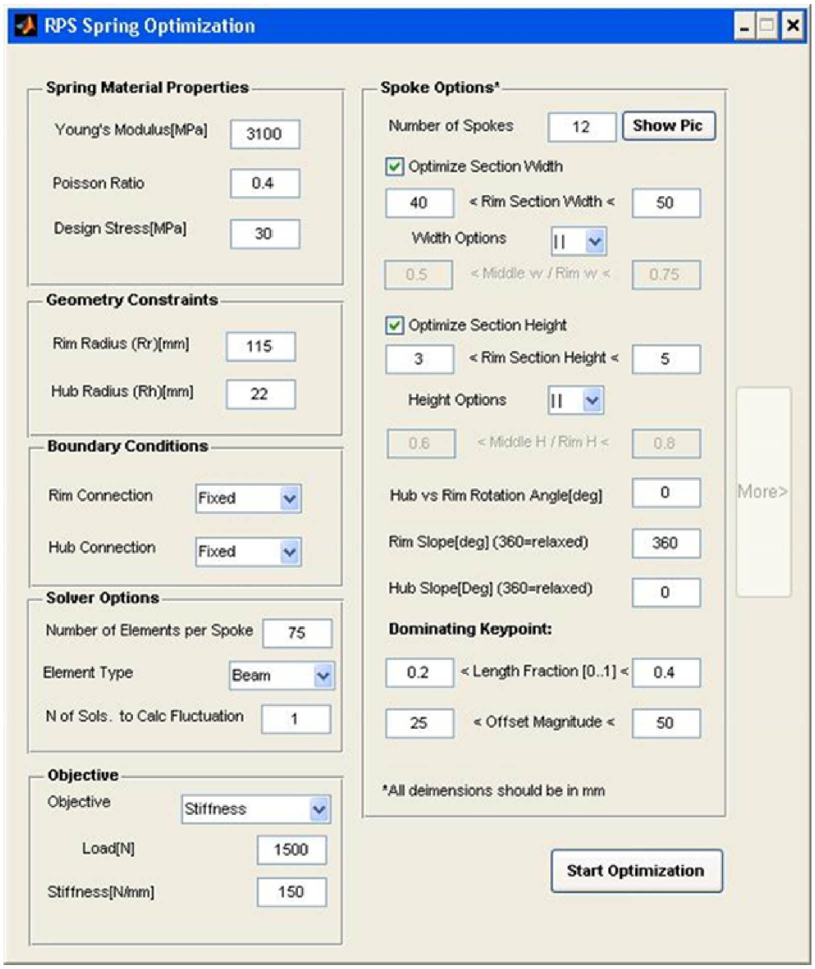
Detailed Review of the Spring Optimization Program

This Appendix presents more information on the spring optimization tool, and how the MATLAB GUI and ANSYS macro communicate. Furthermore, the ANSYS macro which is used by the optimization program to model the springs with Plane elements is described.

The optimization procedure and the task of each the software packages, is shown in Figure A.1. As it is previously described in Chapter 4, the communication between MATLAB and ANSYS is done through text files.

Figure A.1 - The communication between MATLAB and ANSYS

Figure A.2 shows the MATLAB GUI that is used to enter the optimization options. After pressing the "Start Optimization" button, two files are created by MATLAB: "MAT2ANS.DAT" and "PARAMS.DAT". The former includes the optimization parameters and is updated in each iteration. The latter file however, includes the constant parameters which are only created once at the beginning of the optimization. Sample "MAT2ANS.DAT" and "PARAMS.DAT" files are shown in Figure A.3.



Spring Material Properties

Young's Modulus[MPa] 3100

Poisson Ratio 0.4

Design Stress[MPa] 30

Geometry Constraints

Rim Radius (Rr)[mm] 115

Hub Radius (Rh)[mm] 22

Boundary Conditions

Rim Connection Fixed

Hub Connection Fixed

Solver Options

Number of Elements per Spoke 75

Element Type Beam

N of Sols. to Calc Fluctuation 1

Objective

Objective Stiffness

Load[N] 1500

Stiffness[N/mm] 150

Spoke Options*

Number of Spokes 12

☑ Optimize Section Width

40 < Rim Section Width < 50

Width Options ||

0.5 < Middle w / Rim w < 0.75

☑ Optimize Section Height

3 < Rim Section Height < 5

Height Options ||

0.6 < Middle H / Rim H < 0.8

Hub vs Rim Rotation Angle[deg] 0

Rim Slope[deg] (360=relaxed) 360

Hub Slope[Deg] (360=relaxed) 0

Dominating Keypoint:

0.2 < Length Fraction [0..1] < 0.4

25 < Offset Magnitude < 50

*All deimensions should be in mm

Figure A.2 - MATLAB GUI

After creation of the files, ANSYS is run in batch mode in each iteration. The ANSYS macro creates and solves the model using the parameters inputted from the files created by MATLAB. After finishing the solution, ANSYS creates a file named "ANS2MAT.DAT" which contains the solution results (Figure A.3). The last line in the "ANS2MAT.DAT" is the maximum stress; while the previous lines include the stiffness value(s). The number of stiffness values depends on the selected number of solutions defined in MATLAB for monitoring the stiffness fluctuation.

```
Sample  MAT2ANS.DAT

L_R=3.292384e-001
M_V=4.923020e+001
H_RIM=4.538996e+000
H_MID=3.989580e+000
H_R_M=4.404184e+000
H_H_M=4.383327e+000
H_HUB=5.203347e+000
```

```
Sample    PARAMS.DAT

OB_V=0
N_EL=150
N_SPK=6
MAT_E=3100
MAT_V=3.500000e-001
FRC=17
R_R=115
R_H=26
```

```
Sample  ANS2MAT.DAT

 0.19566754E+02
 0.57550361E+02
```

Figure A.3 - Sample text files

Depending on the Elements selected in the MATLAB GUI, the ANSYS code will either use Beam, Shell, or Plane elements for creation of the FEA model. The modeling concept is almost the same. The following pages present the ANSYS macro which is run when a model is created using Plane elements. Each section of the code is described with the text boxes added to the ANSYS code.

```
FINISH
!close the previous open model if any
/CLEAR,
!start an new model
/FILNAME,RPSspring,ON
!set the model title

/PREP7
!start preprocessing
IMMED,0
!turn off automatic graphics update to speed-up the modeling process
```

Start a new session, set the model title and input parameters from Matlab.

```
/input,C:\Docume~1\mazimi\Desktop\RPSPLANE\PARAMS,dat,,,
!input MATLAB file containing material and geometry properties
/input,C:\Docume~1\mazimi\Desktop\RPSPLANE\PASS_S,dat,,,
!input MATLAB file indicating the number of attempts to solve the current
model
/input,C:\Docume~1\mazimi\Desktop\RPSPLANE\MAT2ANS,dat,,,
!input design variables from MATLAB file


!define spring material properties:
MP,EX,1,MAT_E
MP,PRXY,1,MAT_V

!define rigid material properties (used if there is a rev. joint at hub):
MP,EX,2,MAT_E*10
MP,PRXY,2,0


!define element types
ET,1,PLANE42,,,3
ET,2,BEAM3
ET,3,MPC184
```

Plane42 elements are used for meshing the springs Beam3 and MPC184 elements are used for creating revolute joints at hub and rim

```
!put a node at the wheel center if there is a rev. joint at hub
*IF,HUB_J,EQ,1,THEN
 CPT=1
 N,CPT,0,0,0
*ENDIF
```

HUB_J=0: fixed joint at hub

HUB_J=1: revolute joint at hub

RIM_J=0: fixed joint at rim

RIM_J=1: revolute joint at rim

```
R,1,B
!define the width of the plane elements
R,2,B*H_HUB,B*H_HUB**3/12,H_HUB
!add real constant for rigid beams (used in case of rev. joint at hub)


REAL,1
!set the default real constant
```

Real constant 1: width of the plane elements

Real constant 2: section properties of the beam3 elements

```
!define array variables to store key points and lines in:

*DIM,K_1S,ARRAY,6,1
*DIM,K_2S,ARRAY,6,1
*DIM,L_1S,ARRAY,N_EL,1
*DIM,L_2S,ARRAY,N_EL,1
```

K_1S (K_2S): stores key points created above (below) the centerline when assigning section properties.

L_1S (L_2S): stores boundary lines passed through K_1S (K_2S)

```
PHI=PHI*3.14159/180
CPHI=COS(PHI)
SPHI=SIN(PHI)
*IF,SLP_R,NE,360,THEN
   SLP_R=SLP_R*3.14159/180
*ENDIF

*IF,SLP_H,NE,360,THEN
   SLP_H=SLP_H*3.14159/180
*ENDIF
```

Inputted angles from Matlab are converted from degree to radian

SLP_R (SLP_H): end slope at rim (hub)

SLP R (SLP H)= 360: relaxed spline at rim (hub)

```
!define a coordinate system for the model:
 CLOCAL,11,0,,,,,,,
 CSYS,11

!initial variables used in modeling
 SX_LC1=0
 SY_LC1=0
 SX_LC2=0
 SY_LC2=0
 L1SC=0
 L2SC=0
 L_C1=0
 L_C2=0


!Rim connection location:
 X1=R_R
 Y1=0

!hub connection location:
 X3=R_H*CPHI
 Y3=R_H*SPHI
```

X1, Y1: coordinates of key point at rim

X3, Y3: coordinates of key point at hub

X2, Y2: coordinates of key shape controlling key point

```
!dominating point location:
 D13=SQRT((X3-X1)*(X3-X1)+(Y3-Y1)*(Y3-Y1))

 SALPHA=(Y3-Y1)/D13
 CALPHA=(X1-X3)/D13

 X2=X3+L_R*(X1-X3)+M_V*SALPHA
 Y2=Y3+L_R*(Y1-Y3)+M_V*CALPHA
```

```
!create key points:
 *GET,K_MAX,KP,0,NUM,MAX
 K1=K_MAX+1
 K2=K1+1
 K3=K2+1
 KS1=K3+1
 KS3=KS1+1
 K,K1,X1,Y1
 K,K2,X2,Y2
 K,K3,X3,Y3
```

Each key point is represented with an integer value in ANSYS

*GET,K_MAX,… retrieves the highest number assigned to a key point and stores it in K_MAX

Key points are created at specified locations

```
!add rim end slope info and additional key points if necessary:
 *IF,SLP_R,NE,360,THEN
   XV1=COS(SLP_R)
   YV1=-SIN(SLP_R)
    *IF,SLP_R,LT,0,THEN
      XS1=X1+S_R*(X3-X1)
      YS1=Y1+S_R*(Y3-Y1)
      K,KS1,XS1,YS1
    *ELSE
      KS1=0
    *ENDIF

 *ELSE
   KS1=0
 *ENDIF
```

End slopes are converted to unit length vectors

If an end slopes has a negative, an intermediate key point (KS1/KS2) is automatically added to prevent sharp bending of the spring at the ends.

```
!add hub end slope info and additional key points if necessary:
 *IF,SLP_H,NE,360,THEN
   XV3=-COS(PHI+SLP_H)
   YV3=-SIN(PHI+SLP_H)
    *IF,SLP_H,LT,0,THEN
      XS3=X3+S_R*(X1-X3)
      YS3=Y3+S_R*(Y1-Y3)
      K,KS3,XS3,YS3
    *ELSE
      KS3=0
    *ENDIF

 *ELSE
    KS3=0
 *ENDIF

 BSPLIN,K1,KS1,K2,KS3,K3,,XV1,YV1,,XV3,YV3,
 !Pas a Bspline through the selected key points

 *GET,MAIN_L,LINE,0,NUM,MAX
 !Get the created line's number

  *GET,L_LEN,LINE,MAIN_L,LENG
  !Get the created line's length
```

The central spline is drawn with defined key points and end slopes

The integer value representing the line is stored in MAIN_L

```
E_S=(L_LEN/(2*PASS*N_EL))
!approx element size
!(use finer mesh if the solution has not converged at previous attempt(s))


  !find the length fraction of the Bspline's bending point (C2)
   *DO,EL,0,N_EL,1
    FR=(El/N_EL)
    XX=LX(MAIN_L,FR)
    YY=LY(MAIN_L,FR)
    *IF,SQRT((XX-X2)*(XX-X2)+(YY-Y2)*(YY-Y2)),LT,(E_S*PASS),THEN
      FR_DP=FR
      *EXIT
    *ENDIF

   *ENDDO
```

Each point on the centerline can be presented with a length fraction between 0 and 1. The length fraction of the second key point (K2) is required for applying section properties. The value is stored in FR_DP

```
 !in this loop the section height and width are applied to the centerline
and
 !extra key points are created
 !then each 6 consequent points are selected to create boundary lines

 *DO,EL,0,N_EL,1



  FR=(EL/N_EL)
```

Section height is interpolated along the centerline's length. FR is the length fraction and changed from 0 at rim to 1 at hub. H_J is the section height at length fraction FR.

```
     *IF,FR,LT,FR_DP/2,THEN
       H_J=H_RIM+(H_R_M-H_RIM)*(FR/(FR_DP/2))
     *ELSEIF,FR,LT,FR_DP,THEN
       H_J=H_R_M+(H_MID-H_R_M)*((FR-FR_DP/2)/(FR_DP/2))
     *ELSEIF,FR,LT,(1+FR_DP)/2,THEN
       H_J=H_MID+(H_H_M-H_MID)*((FR-FR_DP)/((1-FR_DP)/2))
     *ELSE
       H_J=H_H_M+(H_HUB-H_H_M)*((FR-(1+FR_DP)/2)/((1-FR_DP)/2))
     *ENDIF


    *GET,K_NUM,KP,0,NUM,MAX
    L_C1=L_C1+1
    L_C2=L_C2+1

    K_1=K_NUM+1
    K_2=K_1+1

    SLX=LSX(MAIN_L,FR)
    SLY=LSY(MAIN_L,FR)
```

Key points representing the boundary of the spring are created and stored in the defined arrays (K_1S and K_2S). At the same time boundary lines are drawn by selecting each six consequent boundary key points. The lines are stores in L_1S and L_2S.

```
    K,K_1,LX(MAIN_L,FR)+(-H_J/2*SLY),LY(MAIN_L,FR)+H_J/2*SLX
    K,K_2,LX(MAIN_L,FR)+(H_J/2*SLY),LY(MAIN_L,FR)+(-H_J/2*SLX)

    *IF,L_C1,EQ,7,THEN
     BSPLIN,K_1S(1,1),K_1S(2,1),K_1S(3,1),K_1S(4,1),K_1S(5,1),K_1S(6,1),-
SX_LC1,-SY_LC1
     L1SC=L1SC+1
     *GET,L_1S(L1SC,1),LINE,0,NUM,MAX
     SX_LC1=LSX(L_1S(L1SC,1),1)
     SY_LC1=LSY(L_1S(L1SC,1),1)
     L_C1=2
     K_1S(1,1)=K_1S(6,1)
    *ENDIF

    *IF,L_C2,EQ,7,THEN
     BSPLIN,K_2S(1,1),K_2S(2,1),K_2S(3,1),K_2S(4,1),K_2S(5,1),K_2S(6,1),-
SX_LC2,-SY_LC2
     L2SC=L2SC+1
     *GET,L_2S(L2SC,1),LINE,0,NUM,MAX
     SX_LC2=LSX(L_2S(L2SC,1),1)
     SY_LC2=LSY(L_2S(L2SC,1),1)
     L_C2=2
     K_2S(1,1)=K_2S(6,1)
    *ENDIF

    *IF,L_C1,GT,2,THEN
     *IF,DISTKP(K_1,K_1S(L_C1-1)),LE,0.6*(E_S*PASS),THEN
      KDELE,K_1
      L_C1=L_C1-1
     *ELSE
      K_1S(L_C1,1)=K_1
     *ENDIF
    *ELSE
      K_1S(L_C1,1)=K_1
    *ENDIF

    *IF,L_C2,GT,2,THEN
     *IF,DISTKP(K_2,K_2S(L_C2-1)),LE,0.6*(E_S*PASS),THEN
      KDELE,K_2
      L_C2=L_C2-1
     *ELSE
      K_2S(L_C2,1)=K_2
     *ENDIF
    *ELSE
      K_2S(L_C2,1)=K_2
    *ENDIF

    *IF,L_C1,EQ,1,THEN
     L,K_1,K_2
     *GET,L12,LINE,0,NUM,MAX
    *ENDIF


 *ENDDO
```

The "IF statement" prevents creation of sharp corners when boundary lines are drawn through the boundary key points.

```
 !check to see if there is any key point left. if yes, the remaining
points are
 !selected and boundary lines will be drawn through them:

 *IF,L_C1,NE,2,THEN
   FLST,3,L_C1,3
   *DO,LCI,1,L_C1
    FITEM,3,K_1S(LCI,1)
   *ENDDO
   BSPLIN, ,P51X
   L1SC=L1SC+1
   *GET,L_1S(L1SC,1),LINE,0,NUM,MAX
 *ELSEIF,K_1S(L_C1,1),EQ,K_1,THEN
   BSPLIN,K_1S(1,1),K_1S(2,1),,,,,-SX_LC1,-SY_LC1
   L1SC=L1SC+1
   *GET,L_1S(L1SC,1),LINE,0,NUM,MAX
 *ENDIF
```

If the number of boundary points is not a multiplier of six, the remaining key points are selected and a boundary line is passed through them.

```
 *IF,L_C2,NE,2,THEN
   FLST,3,L_C2,3
   *DO,LCI,1,L_C2
    FITEM,3,K_2S(LCI,1)
   *ENDDO
   BSPLIN, ,P51X
   L2SC=L2SC+1
   *GET,L_2S(L2SC,1),LINE,0,NUM,MAX
 *ELSEIF,K_2S(L_C2,1),EQ,K_2,THEN
   BSPLIN,K_2S(1,1),K_2S(2,1),,,,,-SX_LC2,-SY_LC2
   L2SC=L2SC+1
   *GET,L_2S(L2SC,1),LINE,0,NUM,MAX
 *ENDIF

  L,K_1,K_2
  *GET,L34,LINE,0,NUM,MAX
  L_C1=0
  L_C2=0
```

Draw end lines at hub and rim to close the spring's boundary profile.

```
  !select the drawn boundary lines
   FLST,2,L1SC+L2SC+2,4
   FITEM,2,L12
   *DO,LC,1,L1SC
    FITEM,2,L_1S(LC,1)
   *ENDDO
   FITEM,2,L34
   *DO,LC,1,L2SC
    FITEM,2,L_2S(LC,1)
   *ENDDO

   LSEL,U,LINE,,MAIN_L
   !unselect the centerline
   AL,ALL
   !create an area within the boundary lines
```

Select the drawn boundary lines and unselect the centerline. Then create an area within the selected set of lines (AL,ALL)

```
 !mesh the created area:
 MAT,1
 TYPE,1
 REAL,1

 LESIZE,ALL,E_S
 AMESH,ALL

 ALLSEL,ALL
```

Set the material properties, element type and real constant then mesh the created area.

```
 !generate additional spring using a circular pattern:
 *GET,NLN,NODE,0,NUM,MAX

 CSYS,1
 *IF,N_SPK,GT,1,THEN
   EGEN,N_SPK,NLN,ALL,,,,,,,,,360/N_SPK,
 *ENDIF
```

Generate the rest of the springs using a circular pattern about the center point (EGEN). For the rest of the springs only node and elements are created not the geometry.

```
!apply boundary conditions at rim and hub

!HUB_J         !Hub Joint [0: fixed 1: revolute]
!RIM_J         !Hub Joint [0: fixed 1: revolute]

 *IF,RIM_J,EQ,0,THEN
  CSYS,1
  NSEL,S,LOC,X,0.99*R_R,R_R,
  D,ALL,ALL
  ALLSEL,ALL
 *ELSE


  CSYS,0
  THETA=360/N_SPK
  *DO,NSP,1,N_SPK,1


   LOCAL,11+NSP,0,,,,(NSP-1)*THETA,,,,

   CSYS,11+NSP
   ALLSEL,ALL

   *GET,N1,NODE,0,NUM,MAX
   N1=N1+1
   N,N1,X1,Y1,

   CLOCAL,12+N_SPK,1,X1,Y1,0, , , ,1,1,
   CSYS,12+N_SPK
   NSEL,S,LOC,X,0.9*RJ,RJ,
```

Apply boundary condition at rim and hub.

In case of a fixed joint, first the coordinate system is changed to global cylindrical, and then the nodes adjacent to rim and hub are selected and fixed.

For revolute joints, the coordinate system is set to a cylindrical system, coincident to the joint location. The then nodes within the revolute joint radius (RJ) are selected and connected to the joint center using MPC184 elements. The joint center node is either pinned to the ground (at rim) or connected to the hubs center with rigid beam3 elements (at hub).

Applying boundary conditions continued…

```
  TYPE,3
  *GET,NN,NODE,0,NUM,MIN
  *GET,NH,NODE,0,NUM,MAX
  BB=1
  *DOWHILE,BB
   *IF,NN,EQ,NH,THEN
    BB=0
   *ENDIF
   *IF,NN,NE,N1,THEN
     E,NN,N1
   *ENDIF
   *GET,NN,NODE,NN,NXTH
  *ENDDO

  ALLSEL,ALL
  D,N1,UX,0
  D,N1,UY,0
 *ENDDO
*ENDIF


*IF,HUB_J,EQ,0,THEN
CSYS,1


NSEL,S,LOC,X,0,1.05*R_H,

*GET,NN,NODE,0,NUM,MIN
*GET,NH,NODE,0,NUM,MAX
CPT=NN
*GET,NN,NODE,NN,NXTH
CP,1,UX,CPT,NN
CP,2,UY,CPT,NN
*GET,NN,NODE,NN,NXTH


   BB=1
   *DOWHILE,BB
     *IF,NN,EQ,NH,THEN
       BB=0
     *ENDIF
     *IF,NN,NE,CPT,THEN
       CP,1,UX,NN
       CP,2,UY,NN
     *ENDIF

     *GET,NN,NODE,NN,NXTH
   *ENDDO

*ELSE

 CSYS,0
 THETA=360/N_SPK
```

Applying boundary conditions continued…

```
    *DO,NSP,1,N_SPK,1

     LOCAL,11+NSP,0,,,,(NSP-1)*THETA,,,,

     CSYS,11+NSP
     ALLSEL,ALL
     *GET,N2,NODE,0,NUM,MAX
     N2=N2+1
     N,N2,X3,Y3,
     CLOCAL,12+N_SPK,1,X3,Y3,0, , , ,1,1,
     CSYS,12+N_SPK
     NSEL,S,LOC,X,0.9*RJ,RJ,

     TYPE,3
     *GET,NN,NODE,0,NUM,MIN
     *GET,NH,NODE,0,NUM,MAX
     BB=1
     *DOWHILE,BB
      *IF,NN,EQ,NH,THEN
       BB=0
      *ENDIF
      *IF,NN,NE,N2,THEN
        E,NN,N2
      *ENDIF
      *GET,NN,NODE,NN,NXTH
     *ENDDO
     MAT,2
     TYPE,2
     REAL,2
     E,CPT,N2
    *ENDDO

   *ENDIF


   ALLSEL,ALL
   CSYS,0
```

End of preprocessing.

Solution mode: select static solver and turn nonlinear solver on.

```
 /ESHAPE,1,0
IMMED,1
!turn on automatic graphics update
FINISH
!end of preprocessing

/SOLU
!enter solution mode

!define solver properties:
 ANTYPE,STATIC
 !structural solver
 NLGEOM,ON
 !turn on large deflection solver

 OUTRES,ALL,ALL,,
 !write all sub steps to result file
 TIME,1
 !set the time at the end of the load steps
 NSUBST,4,100,2
 !set the number of load sub steps
```

Set the solvers load step properties: load step starts at t=0 and ends at t=1.

Minimum number of subsets: 2

Maximum number of subsets: 100

Initial number of subsets: 4

```
 !apply displacement/force to hub center
 *IF,OB_V,EQ,1,THEN
  F,CPT,FX,FRC
 *ELSE
  D,CPT,UX,FRC
 *ENDIF

 SOLVE
 !solve model
```

Apply the external load at the hub center node. The load can be either force (OB_V=1) or displacement (OB_V=0).

Solve the simulation and exit solver.

```
FINISH
!exit solver

/POST1
!start post processing

!find displacement and reaction forces at hub center.
 *GET,D_X,NODE,CPT,U,X
  *IF,OB_V,NE,1,THEN
   *GET,R_FX,NODE,CPT,RF,FX
  *ELSE
    R_FX=FRC
  *ENDIF
```

Start post processing.

Get the reaction forces and displacements at hub center.

```
 DEF_F=D_X

 ETABLE,SMAX_I,NMISC,5        !SMAX_I = max. stress at end I of each
                              !element
 ETABLE,SMAX_J,NMISC,10       !SMAX_J = max. stress at end J of each
                              !element
 ETABLE,SMAX_K,NMISC,15       !SMAX_K = max. stress at end K of each
                              !element
 ETABLE,SMAX_L,NMISC,20       !SMAX_L = max. stress at end L of each
                              !element


 !sort the element stress and find the maximum stress:

 ESORT,ETAB,SMAX_I,,1
 *GET,SMAXI,SORT,,MAX
 ESORT,ETAB,SMAX_J,,1
 *GET,SMAXJ,SORT,,MAX
 ESORT,ETAB,SMAX_K,,1
 *GET,SMAXK,SORT,,MAX
 ESORT,ETAB,SMAX_L,,1
 *GET,SMAXL,SORT,,MAX

 SMAX=SMAXI>SMAXJ
 SMAX=SMAXK>SMAX
 SMAX=SMAXL>SMAX


 ALLSEL,ALL


 KK=(R_FX/DEF_F)
 !find the stiffness
 S_MAX=SMAX
 !store the max stress in S_MAX
```

Element tables are used to get stress values for elements. The stress values are then sorted and the maximum stress is stored in SMAX.

KK is the stiffness which is calculated by dividing the reaction force at hub center to its displacement.

At the end, max stress and stiffness values are written to the ANS2MAT text file for Matlab.

```
 FINISH

!export results to MATLAB
*CFOPEN,C:\DOCUME~1\MAZIMI\DESKTOP\RPSPLANE\ANS2MAT,DAT
 FWRITE,KK
 FWRITE,S_MAX
*CFCLOS
```

Appendix B

Detailed Drawings of the 500lb Hand truck Wheel

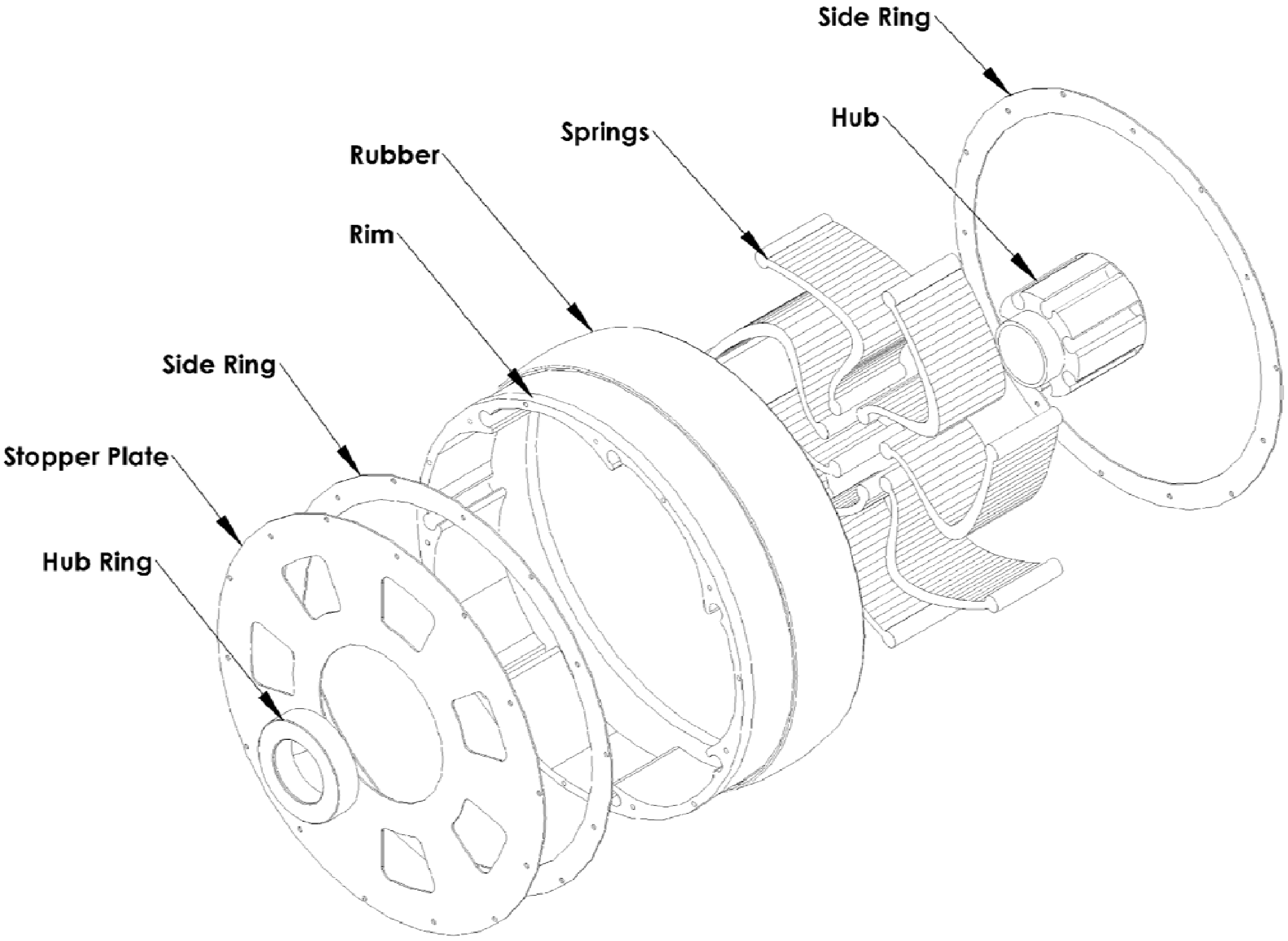
This Appendix Includes:

- Bill of materials for the 500lb hand truck wheel
- Exploded view of the 500lb hand truck wheel
- Detailed drawing of each component of the 500lb hand truck wheel

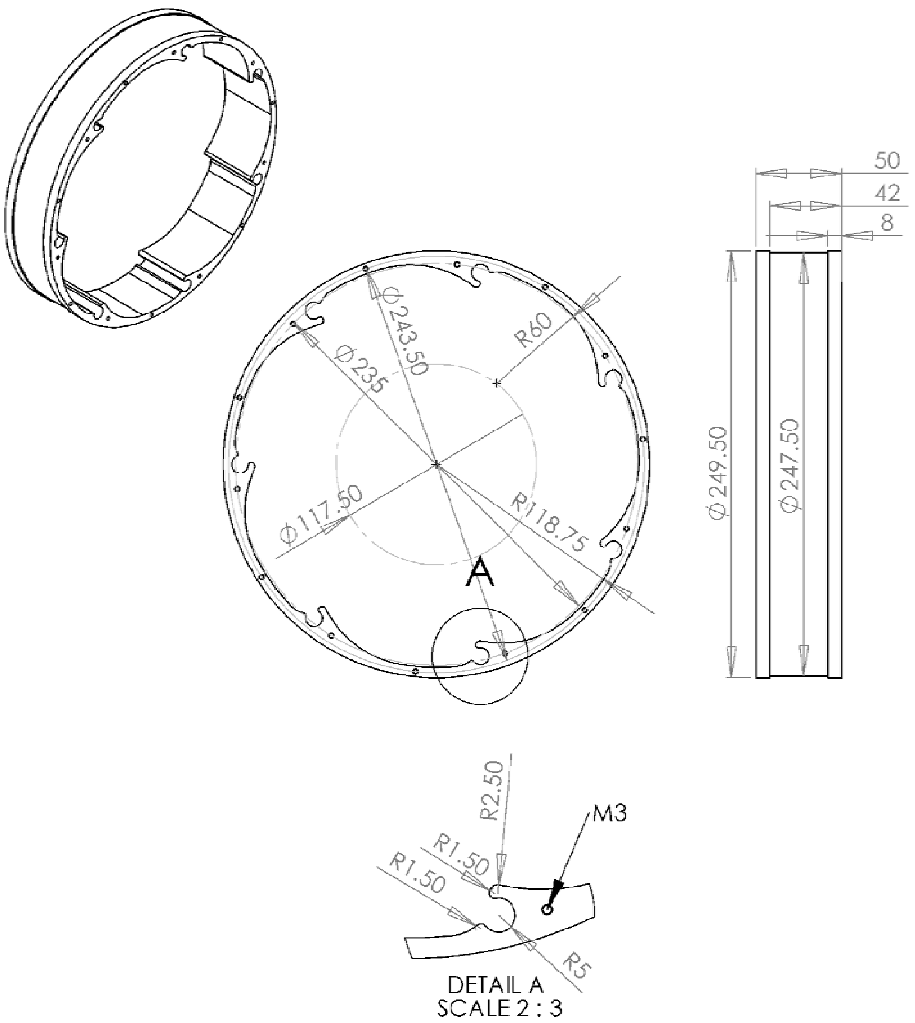
Bill of Material:

Component	**Material**
Rim	Aluminum
Hub	Aluminum
Side ring	Steel
Stopper plate	Steel
Hub ring	Aluminum
Springs	DuPont Delrin 100

Appendix B - Exploded View



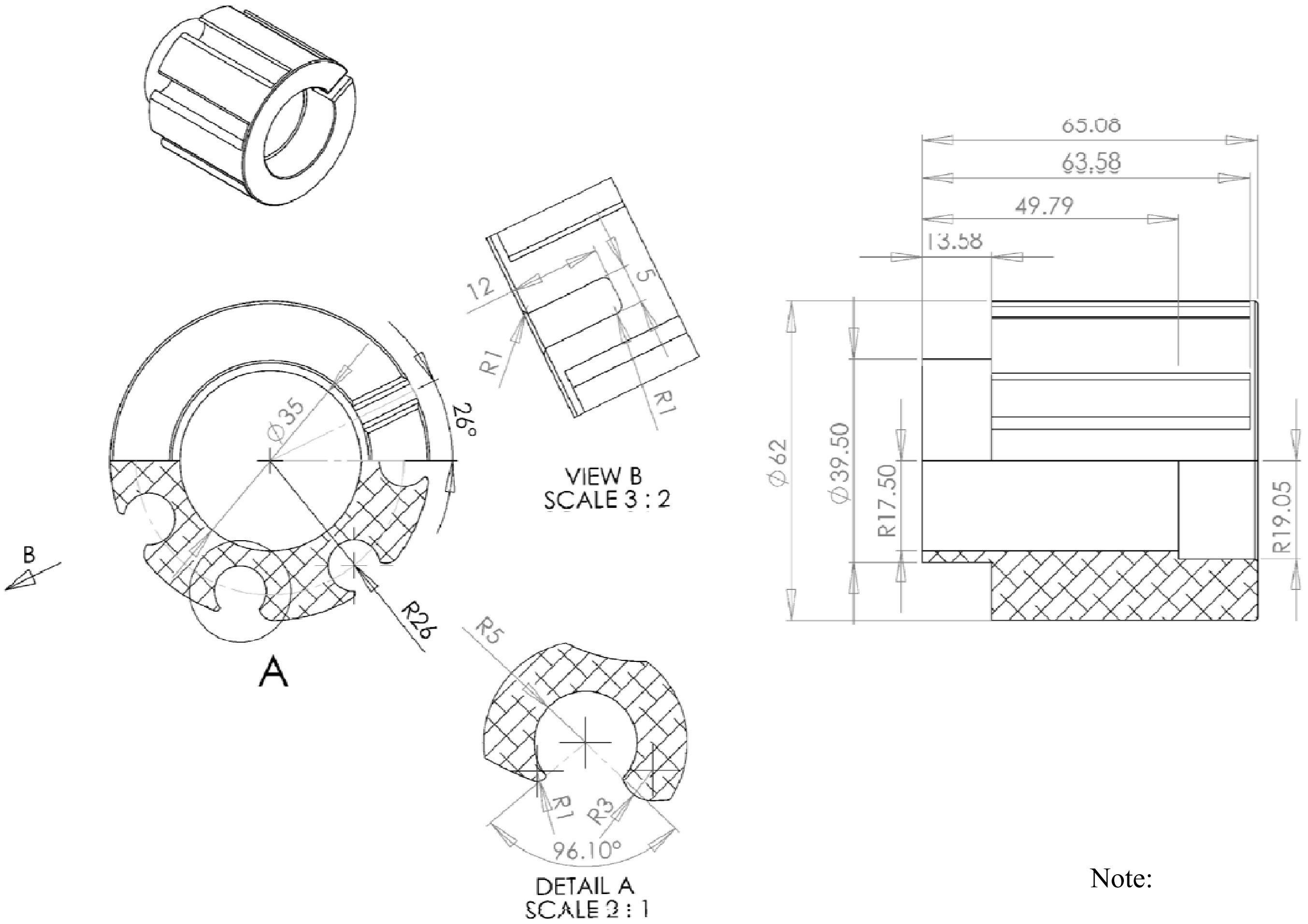
Appendix B - Rim



Note:

All dimensions are in millimeters

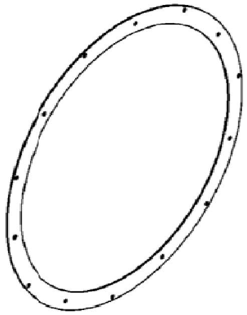
Appendix B - Hub

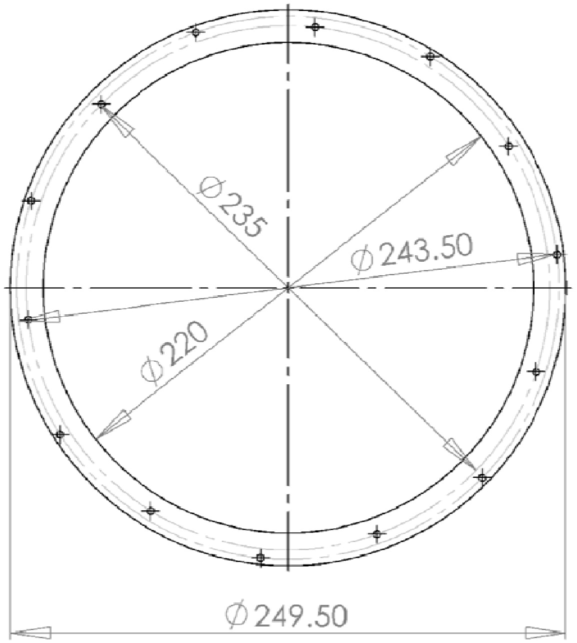


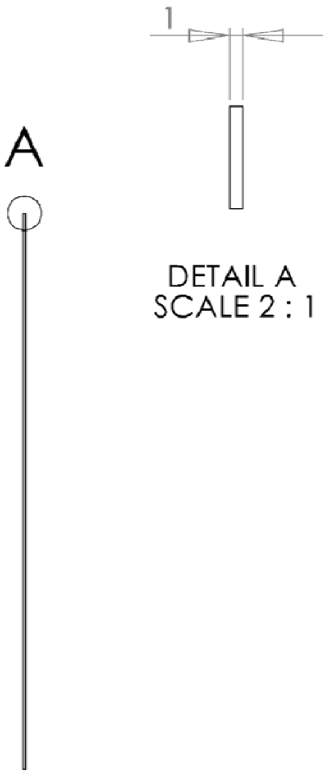
Note:

All dimensions are in millimeters

Appendix B - Side Ring



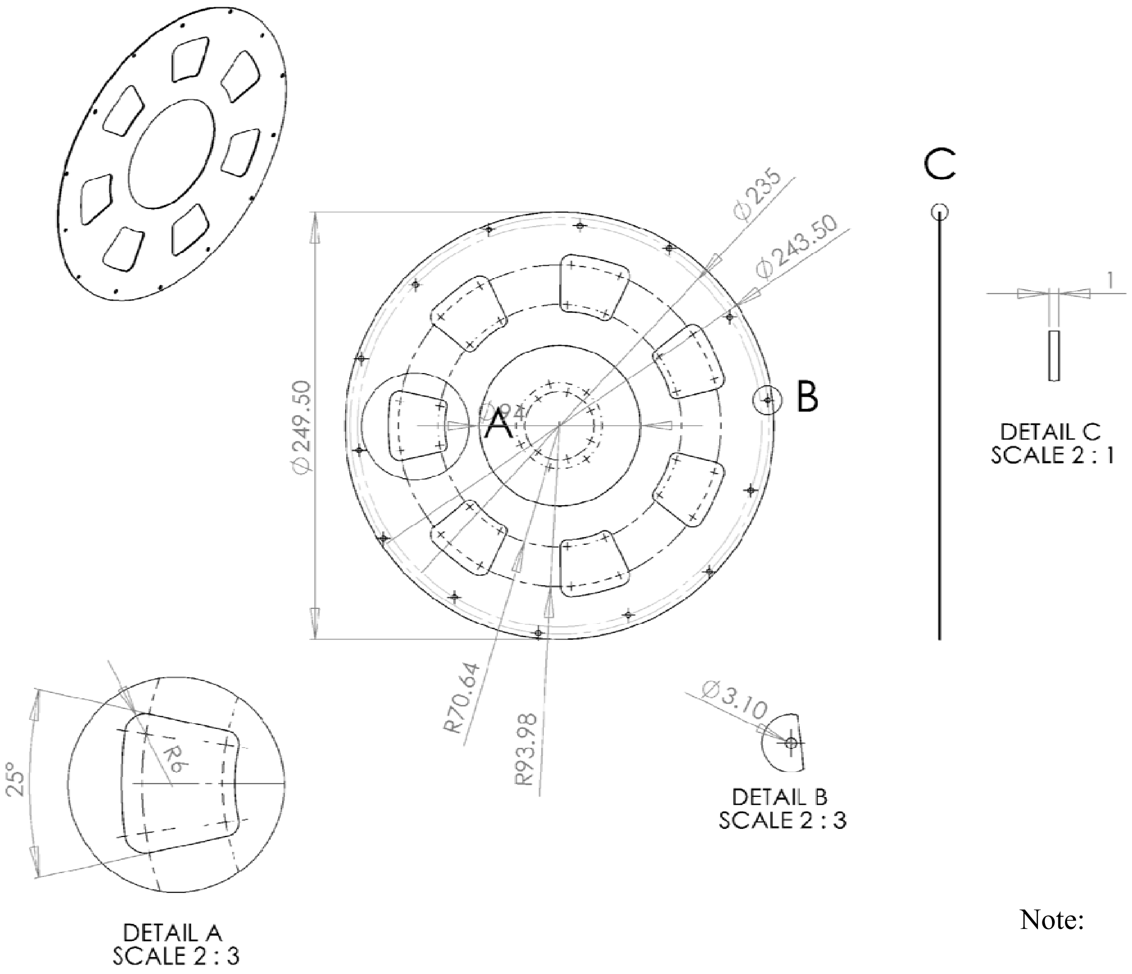




Note:

All dimensions are in millimeters

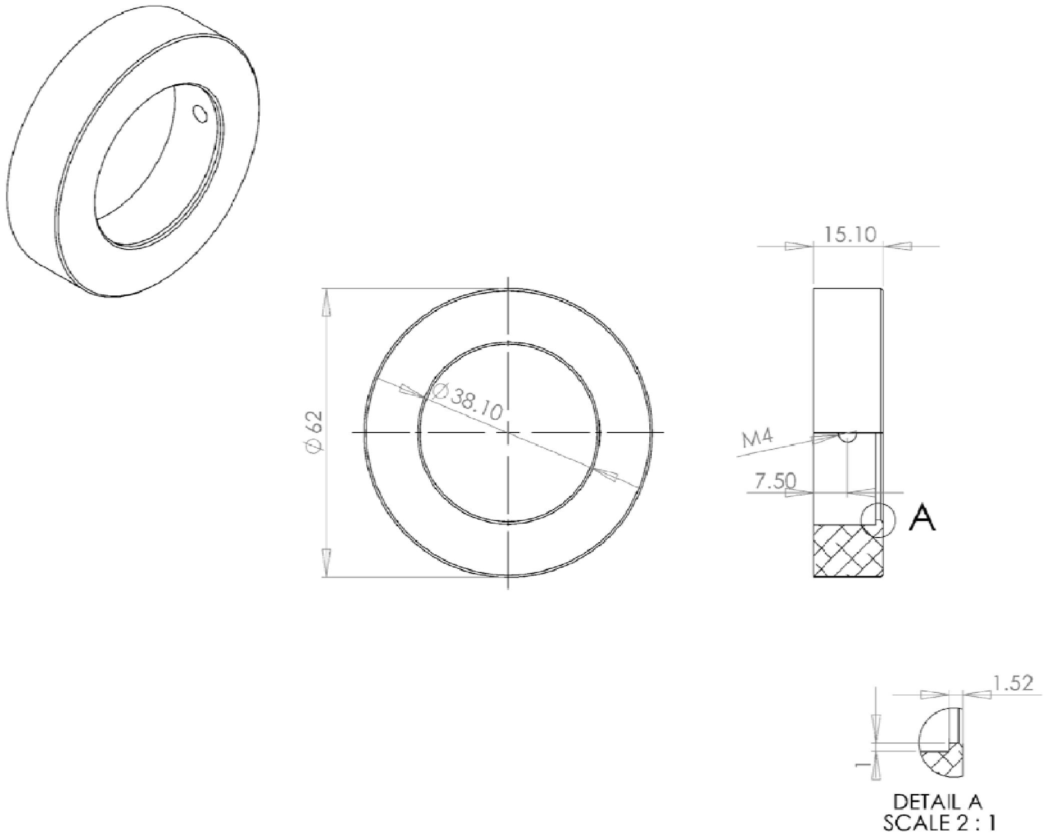
Appendix B - Stopper Plate



Note:

All dimensions are in millimeters

Appendix B - Hub Ring



Note:

All dimensions are in millimeters

Bibliography

National Aeronautics and Space Administration (NASA). Wheels in the Sky. *National Aeronautics and Space Administration (NASA) Web site.* [Online] [Cited: June 1, 2009.] http://marsprogram.jpl.nasa.gov/mer/spotlight/wheels01.html.

A preliminary approach to composite beam design using FEM analysis. **Corvi, A. 1990.** s.l. : Composite Structures, 1990, Vol. 16.

Analysis and optimization of a composite leaf spring. **Shokrieh, Mahmood M and Rezaei, D. 2003.** s.l. : Composite Structures, 2003, Vol. 60, pp. 317-325.

BASF Corporation. 2007. *CAMPUS (Computer Aided Material Preselection by Uniform Standards).* Frankfurt : CWFG (Chemie Wirtschafts Foerderungsgesell schaft GmbH), 2007.

Beer, Ferdinand P and Johnston Jr., E. Russell. 1985. *Mechanics of Materials.* s.l. : McGraw-Hill Inc., 1985. ISBN 0-07-548578-8.

Calculations by Fast Computing Machines Equation of State. **Metropolis, et al. 1953.** s.l. : J. of Chem. Physics, 1953, Vol. 21, pp. 1087-1092.

Composite integrated rear suspension. **Morris, CJ. 1986.** s.l. : Comp Struct, 1986, Vol. 5, pp. 233–242.

Composite leaf springs in heavy truck applications. **Daugherty, RL. 1981.** Tokyo : Composite Materials, Proceedings of Japan–US Conference, 1981.

Composite structures for automobiles. **Beardmore, P. 1986.** 5, s.l. : Comp Struct, 1986, pp. 163–176.

Continental Automotive Systems Inc. Siemense VDO's by-Wire Technology Turns the eCorner. [Online] [Cited: June 11, 2009.] http://usa.vdo.com/press/releases/chassis-and-carbody/2006/SV_20061016_i.htm.

Dixon, John C. 1996. *Tires, Suspension and Handling.* 2nd Edition. s.l. : Society of Automotive Engineers, 1996. ISBN 9781560918318.

Double tapered FRP beam for automotive suspension leaf spring. **Yu, WJ and Kim, HC. 1988.** 9, s.l. : Comp Struct, 1988, pp. 279-300.

DuPont Corporatrion. 2008. *Delrin Design Guide - Module III.* 2008.

Evonik Industries. *High Performance SOLIMIDE® Polyimide Foam Insulation.* Allen, Texas : s.n.

Gillespie, Thomas D. 1992. *Fundamentals of Vehicle Dynamics.* Warrendale : Society of Automotive Engineers, Inc., 1992. ISBN 1560911999.

Ginsberg, J H. 1998. *Advanced engineering dynamics.* 2nd Edition. Cambridge : Cambridge University Press, 1998. ISBN: 0521646049.

Jain, KK and Asthana, R B. 2002. *Automobile Engineering.* London : Tata McGraw-Hill, 2002. ISBN 007044529X.

Khajepour, Amir. 2007. *Design and Prototyping of Reactive Planar Suspension (RPS) Systems for Wheelchairs.* s.l. : University of Waterloo, 2007.

MathWorks, Inc. 2006. *Matlab (R2006b) Help.* s.l. : MathWorks, Inc., 2006.

Michelin Corporation. 2006. Michelin Corporate - News. [Online] April 27, 2006. [Cited: July 1, 2009.] http://www.michelin.com/corporate/actualites/en/actu_affich.jsp?id=17448&lang=EN&codeRubrique=4&actu=true.

Michelin re-invents the wheel. **Kendall, J. 2008.** s.l. : Automotive Engineering International, SAE, 2008, p. 35.

Moins, Stephane. 2002. *Implementation of a simulated annealing algorithm for Matlab.* LINKÖPING : Linköping Institute of Technology, 2002.

Nocedal, J and Wright, S. *Numerical Optimization.* 2000 : Springer. ISBN 0387987932.

Reimpell, J, Stoll, H and Betzler, J W. 2001. *The Automotive Chassis: Engineering Principles.* 2nd Edition. Oxford : Butterworth-Heinemann, 2001. ISBN 0 7506 5054 0.

SAE. 1978. *Vehicle Dynamics Terminology.* Warrendale : Society of Automobile Engineers, 1978.

SAS IP, Inc. 2007. *ANSYS help.* 2007.

Stein, R. 1967. *The Automoblie Book.* s.l. : Paul Hamlyn Ltd, 1967.

The potential for composites in structural automotive applications. **Beardmore, P and Johnson, CF. 1986.** 26, s.l. : Comp Sci Technol, 1986, pp. 251–281.

Using linear graph theory and the principle of orthogonality to model multibody, multi-domain systems. **Schmitke, C and McPhee, J. 2008.** II, Amsterdam : Elsevier Science Publishers B. V., 2008, Vol. 22, pp. 147-160. ISSN:1474-0346.

Van Laarhoven, P J M and Aarts, E H L. 1987. *Simulated Annealing: Theory and Application.* s.l. : Philips Research Laboratories, 1987. ISBN 9027725136.

Why has car-fleet specific fuel consumption not shown any decrease since 1990? Quantitative analysis of Dutch passenger car-fleet specific fuel consumption. **Van den Brink, R and Van Wee, B. 2001.** s.l. : Elsevier Science Ltd, 2001, Transportation Research Part D 6, pp. 75-93.

Yamada, Y, Kuwabara, T and Engineers, Japan Society of Spring. 2007. *Materials for springs.* s.l. : Springer, 2007. ISBN 3540738118.