An Experimentally-Validated V-Belt Model for Axial Force and Efficiency in a Continuously Variable Transmission

Matthew James Messick

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

John M. Kennedy
Alfred L. Wicks
Robert L. West

July 31, 2018
Blacksburg, VA

Keywords: continuously variable transmission, V-belt, Baja SAE

An Experimentally-Validated V-Belt Model for Axial Force
and Efficiency in a Continuously Variable Transmission

Matthew James Messick

Academic Abstract

Rubber V-belt Continuously Variable Transmissions (CVT's) are commonplace in the Baja SAE collegiate design competition, and are also used widely in the power sports industry. These transmissions offer benefits of simplicity in mechanical design, consisting of only two pulleys, and are easy to use. While most teams in the competition use commercially available designs, custom designs are becoming more common, and the Baja team at Virginia Tech has used custom CVT's since the 2014 season. The design of these CVT's has relied heavily on trial and error, requiring significant adjustments to be made during the testing phase. In addition, only qualitative information is known for the relationship between efficiency and design parameters, such as sheave angle. The goal of this thesis is to create an improved V-belt model that may be used as a design tool. This model provides quantitative information about efficiency that can be used to make more informed design decisions. The belt model also provides insight into the magnitude and relationship between the axial forces in the pulleys. This can be used to create an initial design that is more accurate, and possibly reduce the time required for tuning. A CVT dynamometer was constructed to validate the analytical results for efficiency, and this will also serve as a tuning tool for future Baja teams at Virginia Tech. This thesis will advance the state of the custom CVT design and testing process at Virginia Tech, and hopefully lead to improved results at competition in the future.

An Experimentally-Validated V-Belt Model for Axial Force
and Efficiency in a Continuously Variable Transmission

Matthew James Messick

General Audience Abstract

Baja SAE is an annual collegiate competition where students design and build an off-road vehicle. Many teams choose to use Continuously Variable Transmissions (CVT's) in order to maximize the efficiency of the vehicle's transmission. By continually shifting ratios, CVT's allow internal combustion engines to always run at peak performance. There are several types of CVT's, but the most common one used in Baja SAE and the power sports industry is a rubber V-belt design that is controlled mechanically with springs and flyweights. While these devices are used extensively, the underlying dynamics are not well-documented in literature. The Baja team at Virginia Tech builds a custom CVT every year for the vehicle, but the success of the design relies heavily on tuning through trial and error. A better understanding of the dynamics of the belt will result in better initial designs, and will help to reduce the amount of tuning required for the success of the design. This thesis offers an improved dynamic model for a CVT belt, and validates the results of this model through testing on a custom-built dynamometer. This model is also able to predict the efficiency of the transmission, and these results may be used to influence design decisions by predicting their effects on performance. The results of this research will improve the design process for a rubber V-belt CVT and hopefully lead to improved results at competition for the Baja team at Virginia Tech.

Acknowledgements

I would like to especially thank my advisor, Dr. Kennedy, for extending his retirement to see me through my Master's thesis. His help and guidance were invaluable to me during my time as a graduate student. He always kept me pointed in the right direction and on track, and his dedication to the Baja team is very apparent. I would also like to thank the entire 2018 VT Baja team, especially Genevieve Gural, Jay Gonzalez, and Brian Lehman, for providing support and resources to construct the CVT dynamometer. I would like to thank Dewey Spangler and Phil Ratclif for allowing me to work in the Ware Lab and make use of the machine shop. I would like to thank Dr. Wicks and Bhavi Bharat Kotha for allowing me to use their lab and guiding me through the process of designing and assembling printed circuit boards. When I had trouble solving my analytical model in MATLAB, Dr. Lingyuan Kong provided me with the code and solutions from his paper. Without having this as a starting point, I would not have been able to reach a solution for my model. I would also like to thank Dr. Baird and Juan Pretelt for allowing me to use their lab and equipment to measure a belt sample on their DMA. In addition, I would like to thank Doug Harold of the Frith Lab, for allowing me to use his Instron load cell to measure belt properties. Lastly, I would like to thank my parents. Their support through my time as a graduate student has meant more than they know and without them, I would not be where I am today.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Baja SAE Competition

Baja SAE is a collegiate design competition hosted by the Society of Automotive Engineers. Three official competitions are held annually for East, Midwest, and West regions, as well as several other unofficial competitions held at various universities. The goal of the competition is for students to design and fabricate a single-seat off-road vehicle that would serve as a prototype for mass production in a hypothetical company. The competition consists of static events and dynamic events. Static events include a design presentation, sales presentation, and a cost analysis. Dynamic events include a maneuverability course, a suspension course, acceleration runs, and either a hill-climb or sled-pull event. The competition concludes with a four-hour endurance race where the goal is to complete as many laps as possible [1]. Virginia Tech has participated in this competition every year since 1985.

Each team is provided with a 10 horsepower Briggs and Stratton engine. This engine may not be altered in any way as per the rules, but the competition is open to any kind of drivetrain. Many teams purchase continuously variable transmissions (CVT's), and CVTech has a sponsorship program for one of their designs [2]. Some teams create custom designs in an effort to save weight and improve performance. The Baja SAE design team at Virginia Tech has used CVT's on their vehicle since at least 2006 and has been using a custom design since 2014.

1.2 Overview of CVT Operation and Design

Continuously Variable Transmissions (CVT's) are a type of automatic transmission that has an infinite number of possible gear ratios, and this feature offers several advantages over conventional automatic transmissions. Performance characteristics for internal combustion engines vary based on operating speed. All engines have an operating point for which power output or efficiency is maximized. A typical non-dimensional relationship of torque versus rotational speed for an internal combustion engine at wide open throttle is given in [3], and this is plotted in Figure 1.1. The operating point for maximum power output occurs at a higher RPM than for the maximum torque operating point. Conventional automatic transmissions with a finite number of gear ratios force the engine speed to increase with vehicle speed in a given gear ratio, and this causes the engine to operate at speeds other than where it outputs the most power or is the most efficient. A properly designed and tuned CVT will allow the engine speed to remain constant at its optimal operating point, regardless of the vehicle speed.

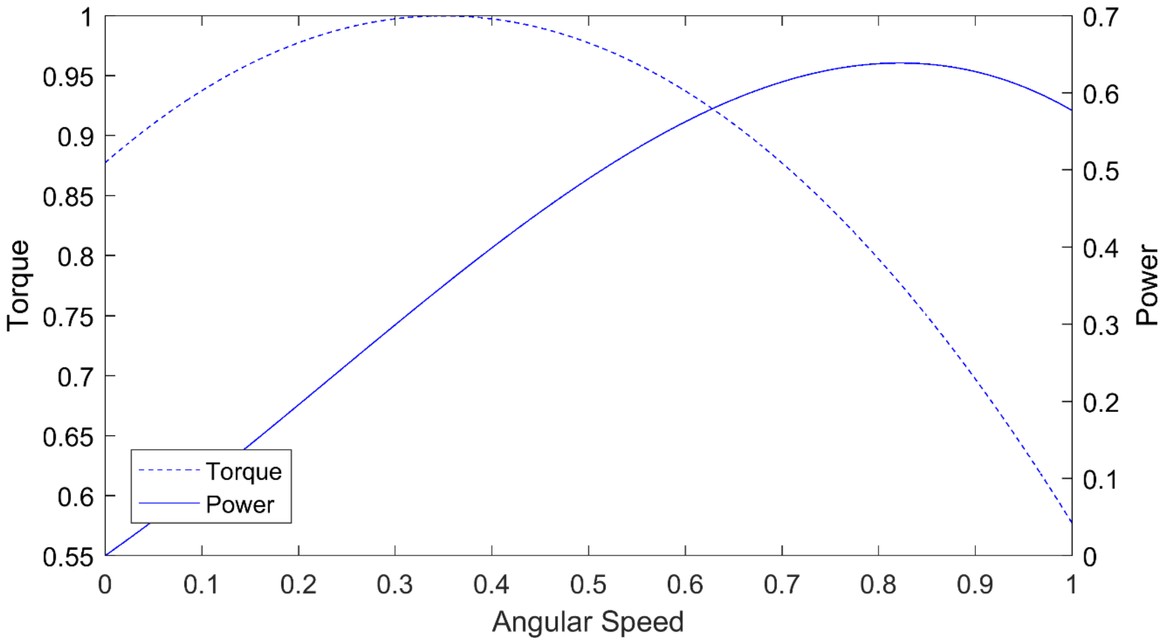


Figure 1.1. Non-dimensional plot of the influence of speed on torque and power for a typical IC engine. The results are normalized to define the angular speed for maximum torque as 0.35, and this results in the maximum power output occurring at approximately ω = 0.82.

Although there are several types of CVT’s in use, this thesis will focus on rubber V-belt designs. This variation has been common in the power sports industry since the 1950’s and is used on vehicles such as all-terrain vehicles (ATV’s), utility terrain vehicles (UTV’s), and snowmobiles [4]. A typical concept of a rubber-belt CVT can be seen in Figure 1.2. The design consists of a drive pulley (primary) that is typically attached to the engine output and a driven pulley (secondary) that is typically mounted on the input to a gearbox. Each pulley consists of two conical sheaves; one is fixed in the axial direction, and the other is allowed to translate axially in order to change the belt radius. This translation may be controlled either electronically or mechanically.

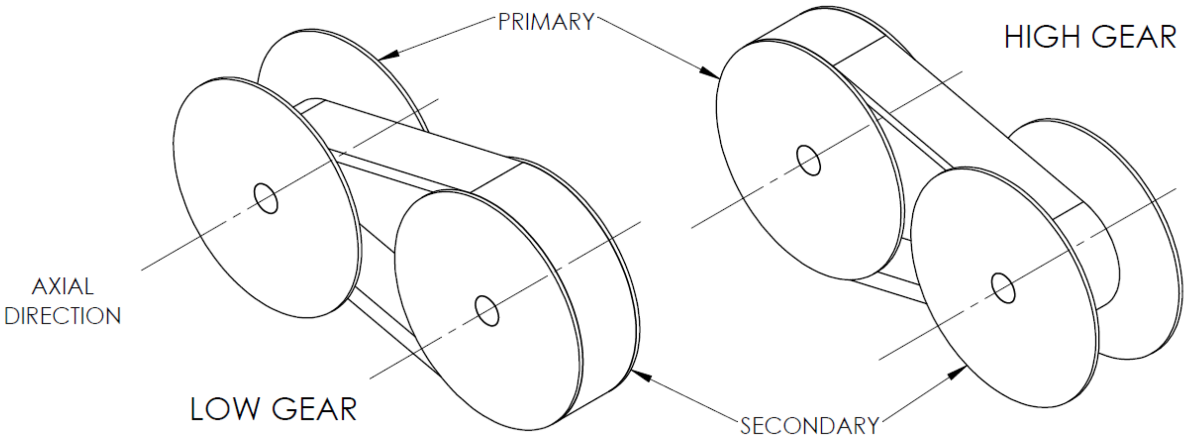


Figure 1.2. Example of a V-Belt CVT. As the sheaves translate on the shaft, the radius of the belt changes. The cases pictured above correspond to a 2:1 (left) and 1:2 (right) reduction ratio.

In a mechanically controlled design, the primary generates an axial force dependent on the rotational speed and the secondary generates an axial force dependent on the torque output of the pulley. As the results of this thesis will demonstrate, these axial forces work together to control the shifting profile of the system, in order to result in a constant engine speed regardless of the load torque experienced on the output shaft. To generate the axial force in the primary pulley, some form of flyweights is used, but the nature of these geometries vary depending on the manufacturer and model. Some typical concepts, including sliding masses and rotating cams, can be seen in Figure 1.3. To generate an axial force dependent on torque, the secondary pulley uses a helical cam design to transmit the torque from the movable sheave to the shaft. This is illustrated in Figure 1.4.

The primary pulley contains a compression spring that is used to counteract the force of the flyweights and keep the sheaves apart when the engine is idling. This enables the CVT to act as a clutch by allowing the belt to slip when the engine is below a certain engagement RPM. The secondary pulley contains a spring that is pre-tensioned in torsion, compression, or both, that provides pressure against the helix and on the belt when the engine is at idle. This also aides in back-shifting to ensure that the system returns to its lowest gear as the vehicle comes to rest.

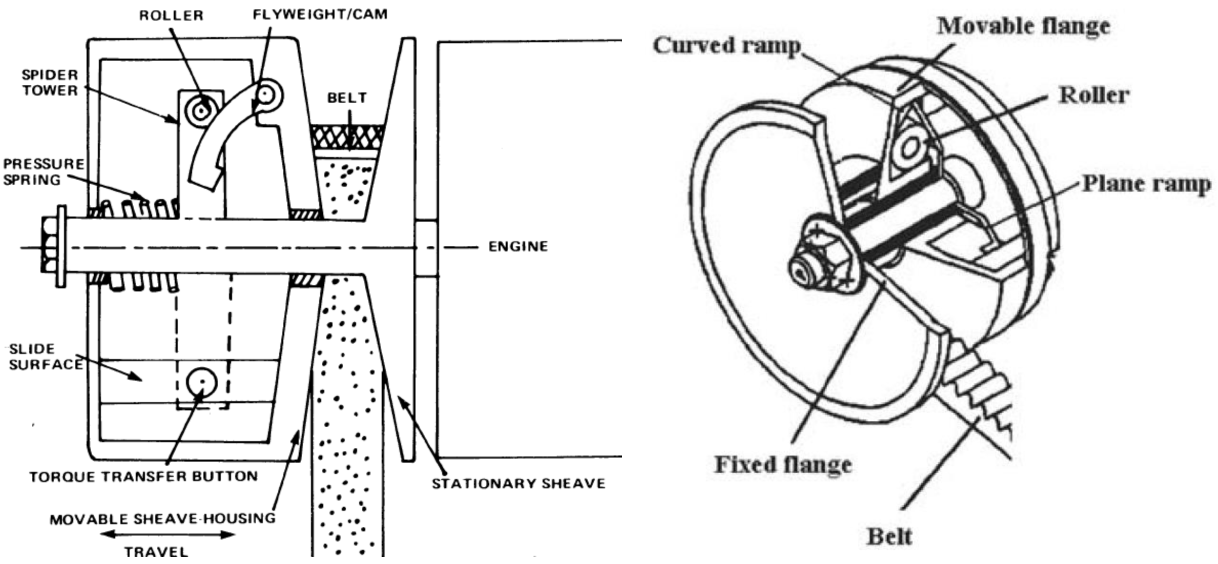


Figure 1.3. Two examples of flyweight designs for the primary. The left depicts a rotating cam/roller concept [4]. The right depicts a sliding mass concept [3].

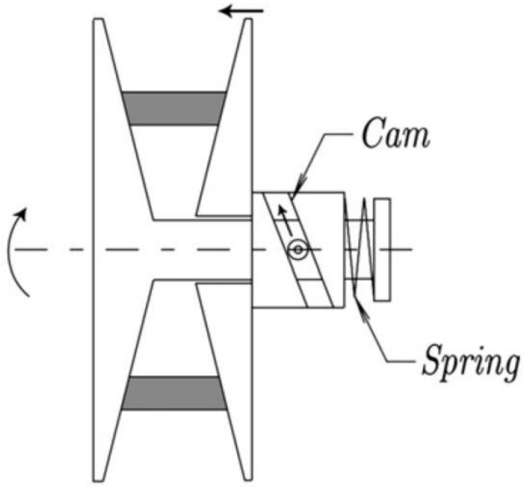


Figure 1.4. Example of secondary helix geometry [5]. A spring provides an axial force when no load is applied and a roller travels along a cam to provide a torque-dependent axial force.

1.3 Goals for this Research

The current process used by the Baja team to design a custom CVT makes numerous approximations in calculating the axial forces exerted by both pulleys and is therefore heavily dependent on tuning to improve the performance of the design. The team does not have a standalone rig for tuning and all tuning must be done on the vehicle by guess-and check. This results in a lengthy tuning process and makes it difficult to reach an optimized design in a reasonable time frame.

This thesis improves the design model used by Baja by analyzing the interactions between the belt and pulley sheaves to better understand the axial forces and how they are related between the primary and secondary pulleys at steady state. This research builds off existing V-belt models in literature [6] to account for the effects of belt bending stiffness and to solve for the entire belt path. The model gives insight into the minimum axial force required in the primary pulley to prevent belt slipping, which aids in the design of the flyweight geometry. The model is also able to calculate the corresponding secondary pulley axial force, and this can be used to design the helix geometry. A properly designed CVT will be able to reach an equilibrium state with the engine running at the desired operating point regardless of the load experienced by the driven pulley. This thesis does not consider shifting response times to a changing load on the driven pulley, and all shifting must be treated as a quasi-equilibrium process. In addition to the axial forces, this thesis considers the effects of certain parameters on efficiency, such as sheave angle, belt properties, and the magnitude of the axial forces. This provides insight for future designs by showing the importance of certain parameters in the design process in order to help prioritize design decisions.

To experimentally validate the results of the model, a standalone dynamometer was constructed to evaluate the design of the commercially available CVT that is currently used as a benchmark for the Baja team. The dynamometer is used to apply a torque load to the secondary pulley and measure the response of the system. By measuring speed and torque at the output and comparing to known engine performance data, the efficiency is determined and compared to analytical data. The construction of the dynamometer also provides a starting point for future studies on transient CVT performance to be conducted.

2 Literature Review

Several authors in literature have addressed the mechanics of V-belt drives, both in the spans and the contact regions in the pulleys. One approach [7] has been to derive and solve governing equations for conservation of momentum of the belt. This results in a system of ordinary differential equations that may be solved as either an initial or boundary value problem, allowing built-in solvers in programs such as MATLAB to be used. This approach is typically used to solve steady-state problems. Other authors have used finite-element methods to solve for the belt path [5]. This approach involves treating the belt as a series of nodes where discretized forms of the momentum equations are applied. More complex models have also used multiple nodes along the belt cross-section to study the contact behavior between the belt and pulley [8]. Transient studies typically employ finite-element methods to study shifting response times for CVT's.

2.1 Application of Flat-Belt Theory to V-Belts

The governing theory behind flat belts is well developed [7]. Early analyses of V-belt drives used this method as an approximation by modifying the friction coefficient. In a flat belt drive, torque is transmitted only through friction acting between the outside surface of the pulley and the inside surface of the belt in the circumferential direction. Reynolds [9] first noticed the presence of slip regions on the pulleys where the belt is stretching or compressing due to its elasticity, which results in unpreventable speed loss in the pulley system. The Capstan Equation [10] relates the maximum tension ratio of a cord wrapped around a cylinder to the contact angle and the friction coefficient:

$$\frac{F_2}{F_1} = e^{\phi\mu} \tag{2.1}$$

where ϕ is the wrap angle and μ is the friction coefficient. Gerbert [7] extends this to a dynamic problem by including inertia effects and redefining the wrap angle as a slip angle:

$$\frac{F_2 - F_c}{F_1 - F_c} = e^{\phi\mu} \qquad F_c = mV^2 \tag{2.2}$$

where m is the belt mass per unit length and V is the belt velocity. The quantity $F - F_c$ is often referred to in literature as the tractive tension, T. Gerbert further explains the existence of an adhesion region where the belt tension remains constant. This occurs where the belt first contacts the pulley on both the drive and driven pulleys, and the length of this region is given as the difference between the known contact angle and the slip angle as calculated in Equation 2.2.

Kong and Parker [11] discuss the additional considerations when bending stiffness is present. Under the Euler-Bernoulli beam equations and conservation of momentum, the bending moment is treated as an equivalent transverse shear term inserted in the momentum equations. However, since the belt radius is equal to the pulley radius wherever the belt is in contact, all derivatives of the moment are zero and the resulting transverse shear term is zero. This allows the above relations for slip and adhesion regions to still hold in the contact regions. If the effects of belt seating are neglected and all friction is assumed circumferential, these equations may be used to describe the mechanics of a V-belt. Gerbert [7] defines a virtual coefficient of friction to be used in this case:

$$\mu_v = \frac{\mu}{sin\beta} \tag{2.3}$$

where β is the sheave angle, or one half of the wedge angle of the belt.

2.2 V-Belt Governing ODE System Methods

For the flat belt approximation of a V-belt, all friction is assumed to be circumferential. When seating is considered, the net friction force on the belt will also have a radial component. Gerbert and Sorge [12] define a sliding angle, γ between the pulley velocity and the belt instantaneous velocity at any given point. In [12], the authors attempt to prove that the presence of an adhesion region is impossible for any one-dimensional model where the belt is treated as having line contact with the pulley. The angle of inclination is defined as the angle between the belt instantaneous radius of curvature and the pulley radius. A positive angle of inclination indicates unseating and a negative angle indicates seating. For a slip region to be present, the angle of inclination must be zero, the derivative of the angle of inclination with respect to arc length must be zero, and the slip angle must be approaching 180 degrees when entering the region. Gerbert and Sorge show that for all of these conditions to be met, any higher derivatives of the above mentioned quantities must also be zero making it impossible for the belt to enter or leave an adhesion region. These conclusions are discussed in further detail in the Model Development and Model Results sections.

The analysis presented in [12] treats the belt as a string with no bending stiffness, and does not consider inertial effects. The analysis only considers the system at steady-state and cannot be used to study shifting. The solution is obtained by solving the governing ordinary differential equation system as an initial value problem and adjusting the initial conditions until a reasonable solution is obtained. Although the authors proved the non-existence of an adhesion region, they adjusted the initial conditions to form an adhesive-like region where the variables remained approximately constant and the angle of inclination is approximately zero.

In [7], Gerbert continues this analysis to include a bending stiffness term. In doing so, he makes several approximations. The pressure of the sheave on the belt is treated as constant over the radial coordinate, and the change of the apparent sheave angle due to the change in sliding angle is neglected. This analysis neglects an inertia term and is also solved as an initial value problem. The initial conditions are varied until reasonable exit conditions are obtained. In both [12] and [7], the drive and driven pulleys are considered individually and the system cannot be modeled as a whole.

Kong and Parker [6] are the first to use available boundary value problem solvers in MATLAB to look at the entire belt-pulley system. This analysis is also steady state and only considers the special case where pulley radii are equal, although the method may be extended to other reduction ratios. By specifying conditions at both the entry and exit points, no iteration is required outside of the solver’s built-in functionality, and no knowledge of the belt behavior within the contact region is required beforehand. In [11], Kong and Parker show that the belt tension would not be constant in a span region where bending stiffness is nonzero due to momentum effects and belt curvature. The model in [6] neglects belt bending stiffness, and this requires the spans to follow a straight line and have constant tension. Boundary conditions are specified such that the entrance and exit slopes for both pulleys form the same line for each span. Kong and Parker mention the difficulty in obtaining solutions outside of the regions presented in [6]. Slack and tight span tensions are specified in the boundary conditions, and as the tight side tension decreases, the derivative of the belt seating variable becomes discontinuous at a point in the driver pulley. This discontinuity causes convergence issues with the boundary value problem solver used, and requires a substantially accurate initial guess for the solution to be provided. The author speculates that including bending effects would result in a smoother solution that would allow the tight side tension to be further decreased, and the initial guess to be less accurate.

2.3 Finite Element and Transient Methods

Several authors have discretized the governing equations to solve for the belt path at a series of nodes. One example is the analysis by Plante and Julio [5], where the shifting behavior of a rubber-belt CVT is modeled. This model accounts for both bending stiffness and damping in bending between the nodes. Friction effects are also treated differently than the previous ODE models. Based on the magnitude of the relative velocity between the belt and sheave, the belt is defined as being either “stuck” or “sliding,” based on a given threshold value. If the belt is “stuck,” the friction force is compared to the maximum possible friction based on a static friction coefficient, and the friction may be less than the maximum possible value in order to match the sum of forces on the node and prevent sliding. As an approximation, this model also assumes that all friction forces are acting in the same plane with no axial component, and the effect of sheave angle on friction is therefore neglected. The model in [5] is experimentally validated using an off-the shelf torque-based driven pulley and a hydraulically actuated drive pulley.

Other models, such as the one presented in [8], consider the distribution of pressure forces against the belt and pulley in the radial direction. The finite-element model contains mesh points along the cross section in both the radial and longitudinal directions. This type of analysis adds complexity that is beyond the scope of this thesis, and is not explored here. A 1-D belt model offers a good trade-off between simplicity and accuracy, making it ideal for use in the design process for a CVT.

3 Model Development

The model presented in this thesis for the belt behavior in the contact regions is similar to the model in [6], with the addition of an equivalent transverse shear term to account for belt bending stiffness and an improved friction model to account for adhesive-like regions along the belt trajectory. Kong and Parker discuss the difficulties in solving their model outside of the region presented in [6]. Adding bending stiffness through the inclusion of a shear term reduces rapid changes in the belt trajectory, and improving the friction model allows the solver to obtain solutions for lower torque and power applications, as seen in Baja SAE. The belt path is solved for as a boundary value problem (BVP). Inclusion of bending stiffness allows the belt spans to have curvature not considered in [6], and the span equations that are derived in [11] are included in this model to solve for the entire belt loop.

3.1 Derivation of Contact Region Governing Equations

Existing string models assume the belt has no bending stiffness, forcing the belt spans to follow a straight trajectory at constant tension. Gerbert derived the equations of motion for a V-belt in [12], but only obtained solutions for each pulley individually by estimating initial conditions. Kong and Parker added inertia effects into Gerbert's equations and solved the equations for both pulleys simultaneously by imposing boundary conditions forcing the belt path to be continuous. This model solves for the entire system simultaneously by also including equations for both span regions in the governing system of differential equations, as taken from [11].

3.1.1 Contact Region Equations as Taken from the Kong and Parker Model

Before considering the forces acting on a differential belt element, the geometry parameters must be defined. Figure 3.1 illustrates the belt differential element and the coordinates that are used to define the belt path. These are the same parameters and relations as used in [12] and [6]. ϕ is the angular coordinate of the belt on the pulley (wrap angle). In the following analysis, ϕ is defined as zero where the belt first contacts the pulley and will increase through the contact region as the independent variable in this analysis. ρ is the instantaneous radius of curvature of the belt. s is the arc length coordinate of the belt. ψ is the angular coordinate of the belt element with respect to the instantaneous belt center of curvature, and $d\psi$ is the angular deformation of the differential belt element. θ is the angle of inclination of the belt element and is defined as the angle between the pulley radius and the belt instantaneous radius of curvature. A positive angle of inclination indicates belt unseating and a negative value indicates belt seating. r is the instantaneous radius of the belt element with respect to the pulley center of rotation. R is the nominal pulley radius, which is the radius the belt would operate at if there was no belt lateral compression and no belt seating in the pulley. Not shown in the figure is the belt radial penetration, which is defined as follows:

$$x = R - r \tag{3.1}$$

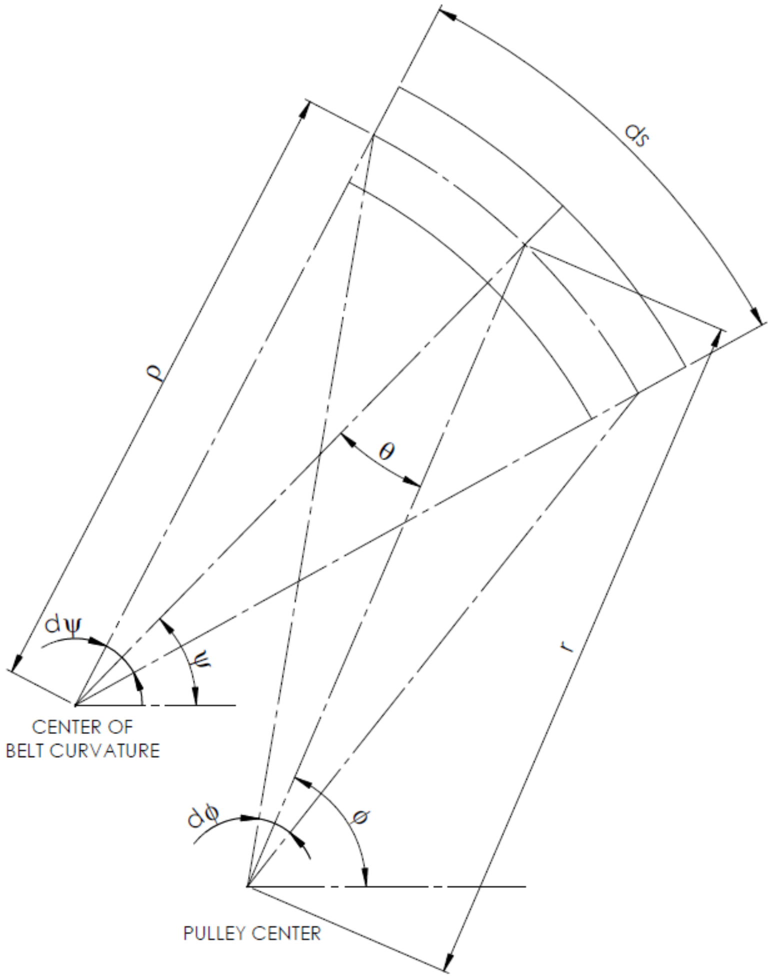


Figure 3.1. Geometry parameters used in the belt model. These parameters relate the belt's trajectory and instantaneous radius of curvature to the pulley geometry.

From Figure 3.1, the following relations arise:

$$\psi = \phi - \theta \qquad \frac{d\psi}{d\phi} = 1 - \frac{d\theta}{d\phi} \tag{3.2}$$

$$ds = \rho\, d\psi \qquad \rho = \frac{ds}{d\psi} \tag{3.3}$$

$$ds = \frac{r\, d\phi}{\cos(\theta)} \tag{3.4}$$

For a system at steady state, the mass flow rate at any point along the belt must be constant. This is defined as G below:

$$G = m(\phi)\, V(\phi) \tag{3.5}$$

where $m(\phi)$ is the belt mass per unit length and $V(\phi)$ is the belt absolute velocity at wrap angle ϕ. The mass per length of the belt when unloaded is defined as m_o and the corresponding velocity if the belt was under no tension is defined as V_o. The only internal force that is considered is the belt tension, F. When deriving the equations for momentum, it is convenient to define the tractive tension as a solution variable, and this quantity is defined in Equation 3.6. The quantity $G\,V$ is a momentum term and has units of N.

$$T = F - G\,V \tag{3.6}$$

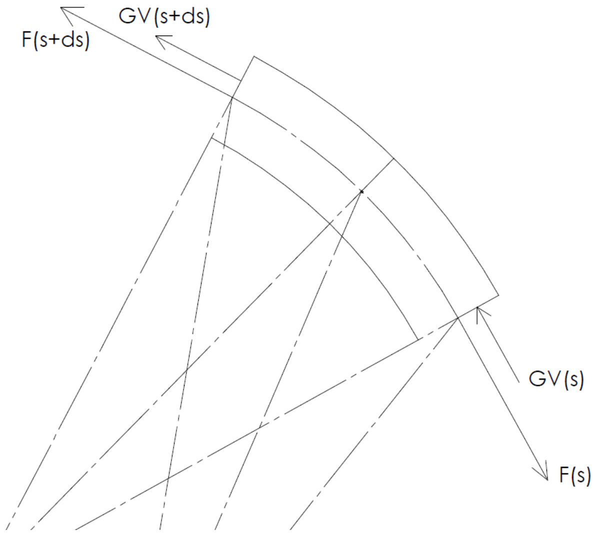


Figure 3.2. Internal forces and momentum terms included in the Kong and Parker model [6].

The external forces acting on the belt are due to the normal forces from the pulley sheaves and the friction from the sheaves. By symmetry, the normal forces from each sheave are equal and the axial components will cancel. The resulting normal force from the sheaves acts on the belt in a direction radially outward from the pulley and its magnitude from each sheave projected in the center plane of the belt is given as follows:

$$F_N = p\ ds \sin(\beta) \tag{3.7}$$

where p is the normal pressure on the belt (force per length) from the sheave. β is the sheave angle (half the belt wedge angle).

To find the force due to friction, the belt element is defined as having a velocity relative to the sheave, V_s. This velocity is illustrated in Figure 3.3. The sheave velocity has magnitude $r\omega$ in the circumferential direction, where ω is the angular velocity in radians per second. The angle of inclination, θ, is used to determine the direction of the belt absolute velocity vector, V. Assuming the belt follows Hooke's law, longitudinal strain is a function of the belt's longitudinal stiffness, EA.

$$\varepsilon = \frac{F}{EA} \tag{3.8}$$

This allows the magnitude of the belt velocity to be determined from the unloaded velocity:

$$V = V_o\left(1 + \frac{F}{EA}\right) \tag{3.9}$$

Using the above equations, a relationship may be determined between the tension and the tractive tension, allowing the belt velocity to be expressed in terms of the tractive tension. The belt unloaded velocity is not known initially, but the belt unloaded mass per length can be measured from a belt sample.

$$F = \frac{\left(T + \frac{G^2}{m_o}\right)}{1 - \frac{G^2}{EA\, m_o}} \tag{3.10}$$

$$V = \frac{G(T + EA)}{EA\, m_o - G^2} \tag{3.11}$$

The relative velocity may then be determined by subtracting the sheave velocity vector from the belt absolute velocity vector. The angle between the relative velocity and the pulley radius is defined as γ and is used to find the direction of the friction force acting on the belt.

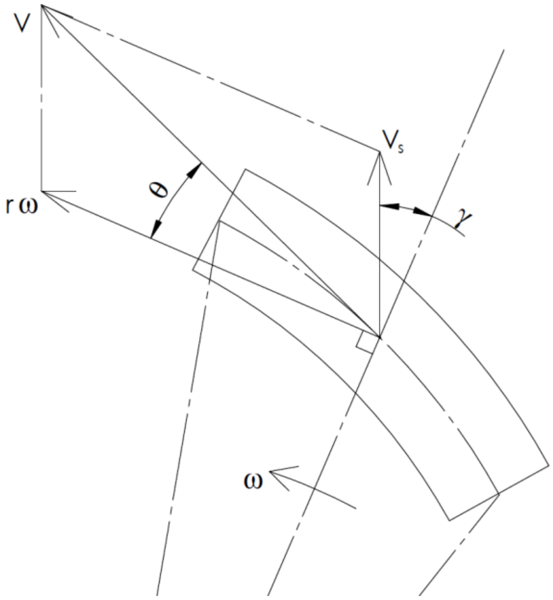


Figure 3.3. Diagram of belt velocity terms. Included are the differential element's absolute velocity (V), sheave absolute velocity ($r\omega$), and relative velocity between the two (V_s).

The radial and circumferential components of the sliding velocity, respectfully, are as follows:

$$\begin{gathered} \mathrm{V}\sin(\theta) \\ V\cos(\theta) - (R - x)\omega \end{gathered} \tag{3.12}$$

By cancelling the absolute velocity using Equation 3.11, the tangent of the sliding angle is calculated as follows:

$$\tan(\gamma) = \frac{\left(\cos(\theta) - \frac{(R - x)\omega(EA\, m_o - G^2)}{G(T + EA)}\right)}{\sin(\theta)} \tag{3.13}$$

The friction acting on the belt element is in a plane tangent to the sheave face. The apparent sheave angle, β_s, is a function of the sheave angle, β, as follows:

$$\tan(\beta_s) = \tan(\beta)\cos(\gamma) \tag{3.14}$$

This angle is measured between the plane that the friction acts in and the center plane of the belt. As an example, if the sliding angle γ is zero, the apparent sheave angle is equal to β. If the sliding angle is 90°, the apparent sheave angle is zero. By symmetry, the friction forces from each sheave are equal, and the axial components cancel. The resulting friction force from each sheave projected in the center plane of the belt is given in Equation 3.15. and the external forces are illustrated in Figure 3.4.

$$\mathrm{F_{FR}} = \mu\, p\, ds \cos(\beta_s) \tag{3.15}$$

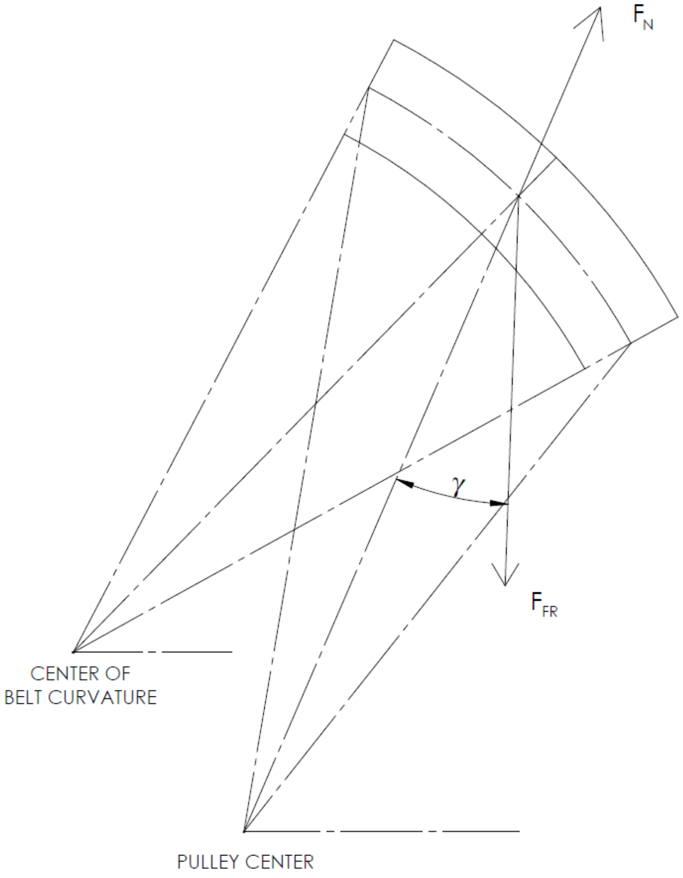


Figure 3.4. Diagram of the external forces acting on the belt differential element. The sheave normal forces act radially outward from the pulley and the friction forces act in a direction opposite the belt relative velocity.

To determine the pressure exerted on the belt from the sheave, the radial penetration along with a radial spring stiffness, k, are used as follows:

$$x = \frac{2\,p_z}{k} \tag{3.16}$$

where p_z is the component of the pressure in the axial direction. Using this relation, the value of k may be determined either experimentally, or through the use of finite element analysis. The pressure, p_z depends both on the axial component of the normal pressure, p, and the axial component of the friction forces, and is determined through the following equation:

$$p_z = p\,(\cos(\beta) + \mu \sin(\beta_s)) \tag{3.17}$$

Combining Equations 3.16 and 3.17 yield an expression for p in terms of x:

$$p = \frac{k\,x}{2(\cos\beta + \mu \sin\beta_s)} \tag{3.18}$$

Conservation of momentum is used to obtain a governing system of ordinary differential equations. The general equation for conservation of momentum states that the sum of forces is equal to the time rate of change of momentum:

$$\sum F = \frac{d}{dt}(mv) \tag{3.19}$$

By using the small angle approximations for $\sin(\psi)$ and $\cos(\psi)$, the time rate of change of momentum for the belt element in the normal direction is $G\,V\,d\psi$ and the time rate of change of momentum in the tangential direction is $d(G\,V)$. Summing the belt forces in the normal and tangential directions, respectively, yield the following equations:

$$G\,V\,d\psi = F\,d\psi - 2p\,(\sin\beta\cos\theta - \,\mu\cos\beta_s\cos(\theta+\gamma))\,ds \tag{3.20}$$

$$d(G\,V) = dF - 2p\,(-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma))\,ds \tag{3.21}$$

Converting the belt arc length, ds, to wrap angle, $d\phi$, with Equation 3.4, replacing the tension with tractive tension, and replacing the belt angular deflection, $d\psi$, with the expressions from Equation 3.3, yield the following two first-order differential equations:

$$\theta' = 1 - \frac{2p}{T}\left(\sin\beta\cos\theta - \mu\cos\beta_s\cos(\theta+\gamma)\right)\frac{R-x}{\cos\theta} \tag{3.22}$$

$$T' = 2p\left(-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma)\right)\frac{R-x}{\cos\theta} \tag{3.23}$$

where the derivatives are with respect to ϕ. The remaining equation for x may be determined by inspection from the geometry:

$$x' = -(R-x) * \tan(\theta) \tag{3.24}$$

where G is an unknown constant, and is determined through a fourth equation:

$$G' = 0 \tag{3.25}$$

Equations 3.22-3.25 constitute the system of four first order, coupled ordinary differential equations as derived in [9] and [6].

3.1.2 Additional Effects taken into Consideration

Kong and Parker mention the difficulties in solving the system of equations outside of the range of parameters presented in [6]. Their results show rapid changes in the belt trajectory that arise as the belt tension decreases, and this results in discontinuities in the derivatives. They speculate that including belt bending stiffness in the analysis would have the effect of smoothing the belt path, especially in the entrance and exit regions, and this would have the effect of improving the solution convergence. To include this effect, a method similar to the one presented in [11] is used. The belt has a bending stiffness, EI, and the internal moment can be determined using Euler-Bernoulli beam theory:

$$M = \frac{EI}{\rho} = EI\,\frac{d\psi}{ds} \tag{3.26}$$

Treating the moment, M, as an equivalent transverse shear allows application to the momentum equations. Kong and Parker in [11] show that conservation of angular momentum yields the expression in Equation 3.27 for transverse shear. The belt moment and shear terms are shown in Figure 3.5.

$$Q = \frac{dM}{ds} = EI\ \frac{d^2\psi}{ds^2} \tag{3.27}$$

The sum of these shear forces act in both the normal and tangential directions, and are added to Equations 3.20 and 3.21 as follows:

$$G\ V\ d\psi = F\ d\psi - 2p\ (\sin\beta\cos\theta - \ \mu\cos\beta_s\cos(\theta+\gamma))\ ds + dQ \tag{3.28}$$

$$d(G\ V) = dF - 2p\ (-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma))\ ds + Q\ d\psi \tag{3.29}$$

Rearranging as before using the relations in Equations 3.2-3.4 yield the following equations:

$$\theta' = 1 - \frac{2p}{T}\ (\sin\beta\cos\theta - \ \mu\cos\beta_s\cos(\theta+\gamma))\ \frac{R-x}{\cos\theta} + \frac{1}{T}Q' \tag{3.30}$$

$$T' = 2p\ (-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma))\ \frac{R-x}{\cos\theta} - Q + Q\ \theta' \tag{3.31}$$

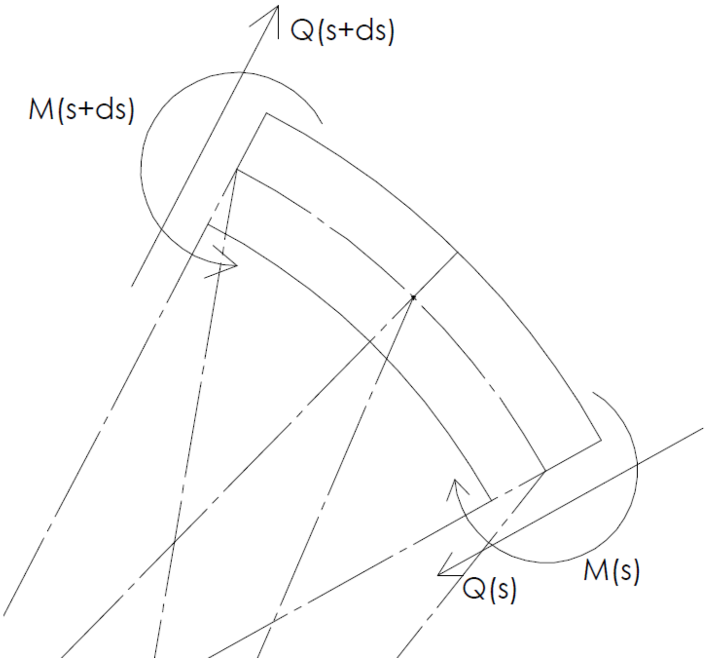


Figure 3.5. Diagram of the shear and moment terms experienced by the belt differential element.

To constitute a coupled ODE system, the shear must be expressed in terms of the solution variables, T, θ, and x. Equations 3.2 and 3.4 are first used to transform the moment in Equation 3.26 to a function of θ and x:

$$M = \frac{\cos(\theta)}{R-x}\, EI\,(1-\theta') \tag{3.32}$$

Equation 3.2 is then used to convert the derivative of M with respect to s from Equation 3.27 into a derivative with respect to ϕ:

$$Q = \frac{dM}{ds} = \frac{\cos(\theta)}{R-x}\, M' \tag{3.33}$$

$$M' = \frac{EI\cos(\theta)\, x'}{(R-x)^2} - \frac{EI\cos(\theta)\, x'\theta'}{(R-x)^2} - \frac{EI\cos(\theta)\, \theta''}{R-x} - \frac{EI\sin(\theta)\, \theta'}{R-x} + \frac{EI\sin(\theta)\, (\theta')^2}{R-x} \tag{3.34}$$

Similarly, the first derivative of Q with respect to ϕ from Equation 3.30 may also be determined:

$$Q' = \frac{-\sin(\theta)\, \theta'(R-x) + \cos(\theta)\, x'}{(R-x)^2}\, M' + \frac{\cos(\theta)}{R-x} M'' \tag{3.35}$$

$$\begin{aligned} M'' = {} & 2\frac{EI\cos(\theta)\, (x')^2}{(R-x)^3} - 2\frac{EI\cos(\theta)\, (x')^2\theta'}{(R-x)^3} - \frac{EI\cos(\theta)\, (\theta')^2}{R-x} + \frac{EI\cos(\theta)\, (\theta')^3}{R-x} \\ & + \frac{EI\cos(\theta)\, x''}{(R-x)^2} - \frac{EI\cos(\theta)\, \theta' x''}{(R-x)^2} - 2\frac{EI\cos(\theta)\, x'\theta''}{(R-x)^2} - \frac{EI\cos(\theta)\, \theta'''}{R-x} \\ & - 2\frac{EI\sin(\theta)\, x'\theta'}{(R-x)^2} + 2\frac{EI\sin(\theta)\, x'(\theta')^2}{(R-x)^2} - \frac{EI\sin(\theta)\, \theta''}{R-x} \\ & + 3\frac{EI\sin(\theta)\, \theta'\theta''}{R-x} \end{aligned} \tag{3.36}$$

The results presented in [6] indicate that the belt radial penetration is small compared to the nominal radius. In addition, when the belt tension is relatively low, the angle of inclination is also small. These values are the largest in the entrance and exit regions, and their effects are lessened with the inclusion of bending stiffness, justifying linearization of the expressions for shear and its derivative as an approximation around the following values:

$$\theta = \theta' = \theta'' = x = x' = x'' = 0 \tag{3.37}$$

To find a linear expression for Q, Mathematica is used to substitute Equation 3.34 into Equation 3.33 and evaluate partial derivatives with respect to the variables being linearized. This yields the following approximate expression for Q, which is substituted into Equation 3.31 and shown in Equation 3.39.

$$Q \approx -\frac{EI}{R^2}\theta'' + \frac{EI}{R^3}x' \tag{3.38}$$

$$T' = 2p\,(-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma))\;\frac{R-x}{\cos\theta} + (\theta' - 1)\left(-\frac{EI}{R^2}\theta'' + \frac{EI}{R^3}x'\right) \tag{3.39}$$

Similarly, Mathematica is used to substitute Equations 3.34 and 3.36 into Equation 3.35 and take partial derivatives to find an approximate expression for Q':

$$Q' \approx \frac{EI}{R^3}x'' - \frac{EI}{R^3}(R + 2\,x)\theta''' \tag{3.40}$$

This expression contains a third derivative of θ. Since this is the highest order derivative, Equation 3.30 must be rearranged in terms of this value to convert the system to first order, as required by the boundary value problem solver. This is shown in Equation 3.41.

$$\theta''' = \frac{TR^3}{EI(R+2\,x)}\left(1 - \frac{2p}{T}\;(\sin\beta\cos\theta - \;\mu\cos\beta_s\cos(\theta+\gamma))\;\frac{R-x}{\cos\theta} - \theta' + \frac{1}{T}\frac{EI}{R^3}x''\right) \tag{3.41}$$

An expression for x'' is obtained by deriving Equation 3.24:

$$x'' = -\frac{(R-x)\theta'}{\cos(\theta)^2} + \tan(\theta)\,x' \tag{3.42}$$

The governing ODE system for the belt in the contact region with bending stiffness is comprised of Equations 3.39, 3.41, 3.42, and 3.25.

To further extend the range of parameters for which a solution may be obtained, an improved friction model was also added. Gerbert and Sorge attempted to prove mathematically the non-existence of an adhesion region within the belt contact region in [12], and Kong and Parker in [6] use this assumption by assuming the belt is always experiencing dynamic friction. Other models have attempted to account for adhesive regions. The finite element model in [5] uses a friction model developed by Karnopp [13]. This model defines a threshold slip velocity, below which it is assumed that the belt is in adhesion. In an adhesion region, a static friction coefficient is used and it is assumed that the force due to friction is equal and opposite to the sum of all other forces acting on the belt.

As the magnitude of the sliding velocity vector goes to zero, Equation 3.13 fails to calculate the sliding angle. Implementing a Karnopp friction model would be tedious for a differential belt model. Alternate differential equations would need to be derived to govern the belt motion in an adhesion region. The friction model that is used in this analysis is a viscous Coulomb model . This model also incorporates a threshold slip velocity, but below this value the coefficient of friction is decreased linearly. This relation is illustrated in Figure 3.6. This approximation prevents a true “adhesion” region from occurring, and this forces the sliding velocity to be non-zero. This allows the sliding angle, γ, to still be calculated, and allows the derived equations to be solved without further modifications.

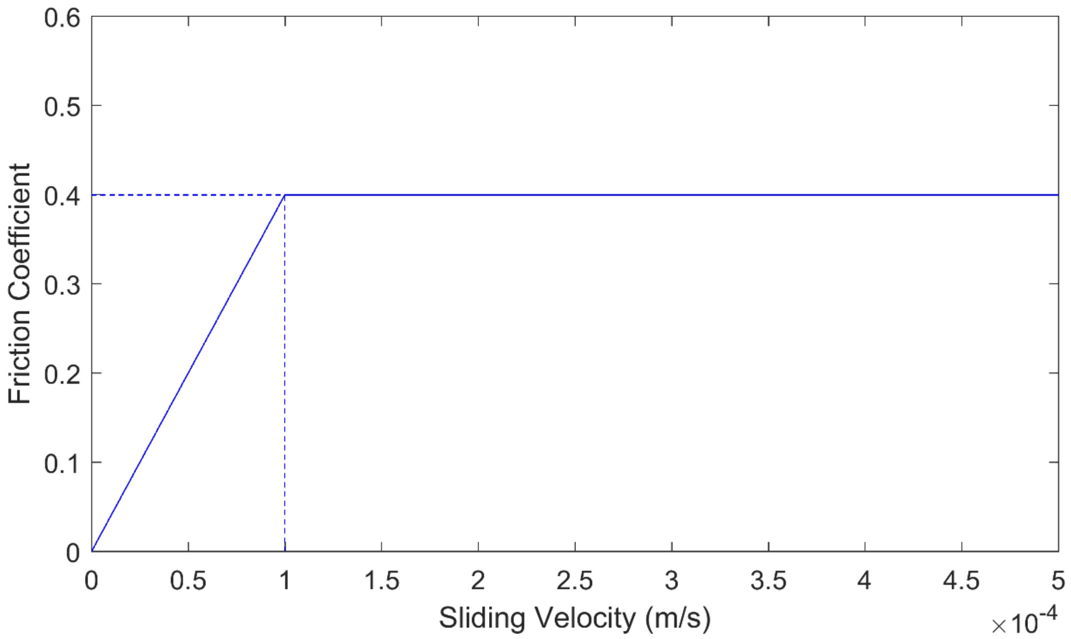


Figure 3.6. Plot of the friction model used in this analysis. A threshold velocity is defined to separate the adhesion and sliding regions, shown above as 0.0001m/s. Below this relative velocity, the friction coefficient is decreased linearly from the dynamic value of 0.4 to 0.

3.2 Derivation of Span Governing Equations (from [8])

Similar to the equations for the contact region, the governing equations for the belt span are equations for the conservation of momentum in the belt tangential and normal directions. Unlike the contact region, the belt experiences no external forces in this region. The only forces are internal due to tension and transverse shear, representing the bending stiffness. The equations of motion for a belt with bending stiffness in the spans have already been developed by Kong and Parker [11]. These equations are summarized in this section. Figure 3.7 illustrates the free-body diagram of a belt differential element in the span. In the spans, ϑ denotes the angle of inclination of the belt with respect to the horizontal. Similar to the contact region, the tractive tension is used to include both tension and momentum effects in a single term. For all span equations, the tractive tension is denoted as τ. Using Euler-Bernoulli beam theory, the shear force due to bending is expressed below:

$$Q = \frac{dM}{ds} = EI\frac{d^2\vartheta}{ds^2} \tag{3.43}$$

Summing the forces in the tangential direction leads to the following equation for momentum:

$$dF = G\ dV - Q\ d\vartheta \tag{3.44}$$

$$\tau' + EI\ \vartheta'\ \vartheta'' = 0 \tag{3.45}$$

Summing the forces in the normal direction leads to the following equation for momentum:

$$(F - GV)\ d\vartheta - dQ = 0 \tag{3.47}$$

$$\tau\ \vartheta' - EI\ \vartheta''' = 0 \tag{3.48}$$

All derivatives in the span equations are with respect to the arc length coordinate, s. Equations 3.47 and 3.48 constitute the ODE system as derived in [11].

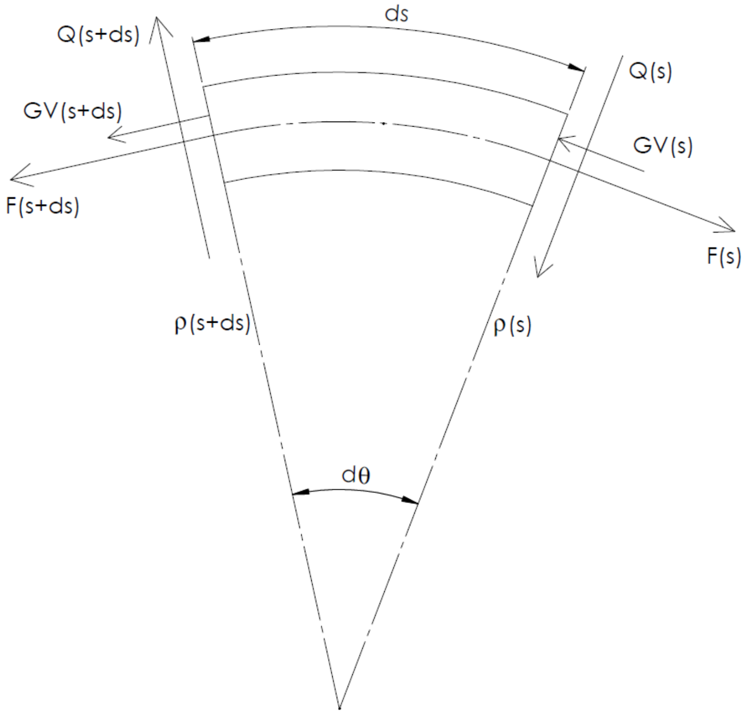


Figure 3.7. Free body diagram of the belt differential element in the span regions. The only forces are internal, from tension and shear due to bending.

4 Belt Testing

The belt properties used for this analysis are obtained from a CVTech belt, model number BD52-2172-S. This is one of the options given as part of the sponsorship package offered by CVTech for the Baja SAE competition [2]. The belt is rubber, with aramid fiber reinforcements and fabric on the outer and inner surfaces to provide resistance to abrasion. It is a single cog design, with teeth on the inside belt surface only. A CAD model of the belt is shown in Figure 4.1.

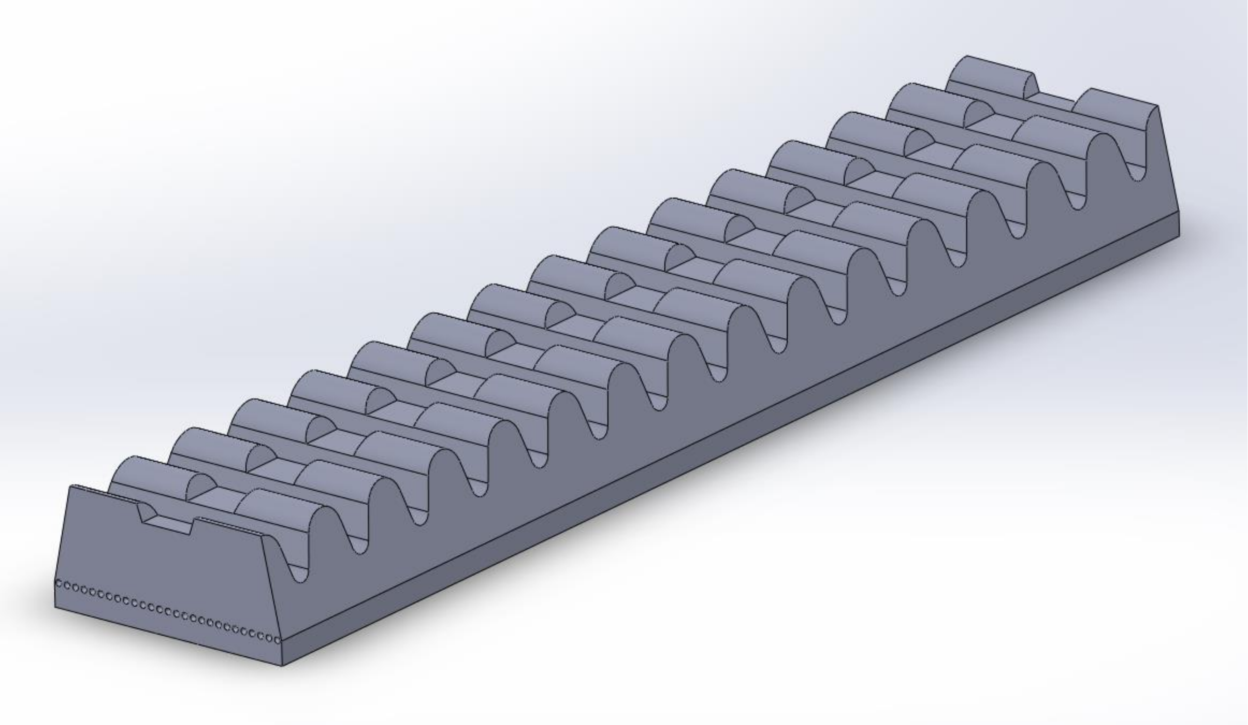


Figure 4.1. CAD model showing a section of the CVTech belt used in this analysis. Teeth are located on the inside surface of the belt only and the cross section contains 27 aramid fibers for reinforcement in the longitudinal direction.

4.1 Measured Rubber Properties

To determine the mechanical properties of the belt, the rubber and fiber reinforcements are treated separately. The properties of the rubber are determined using a Dynamic Mechanical Analyzer (DMA). This instrument allows loads to be applied at high frequency and high strain rates, representing the loading seen when in assembly. Rubber is a polymer exhibiting both elastic and viscous properties. Static testing on a load cell is only capable of determining the elastic properties of the rubber. A DMA is capable of quantifying both the elastic and viscous properties of the material in order to evaluate the validity of an elastic-only belt model. The Ares G2 DMA that is used to obtain the properties is shown in Figure 4.2.

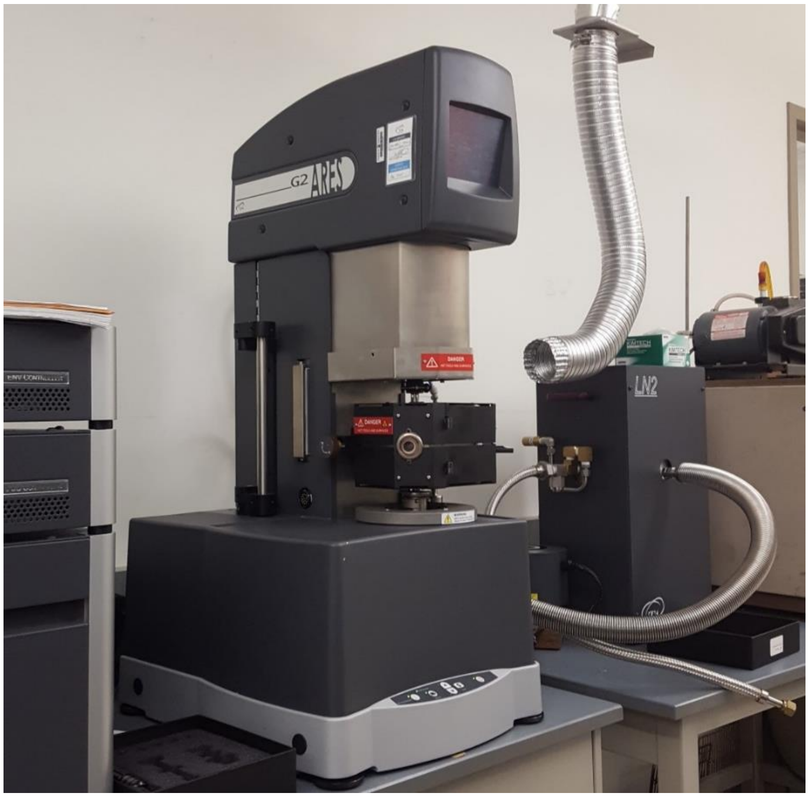


Figure 4.2. Dynamic Mechanical Analyzer used to obtain the rubber properties of the CVTech belt. An oven around the sample allows the temperature dependence of the material properties to be determined.

The rubber is assumed to be isotropic, so the stiffness properties in any direction would be the same and the orientation of the obtained sample would not affect the results. A rectangular sample is obtained from one of the teeth on the belt. The cross section of the sample is 5.46mm x 8.20mm, and the vertical distance between the clamps is 13.22mm. The sample is tested in torsion at frequencies ranging between 10Hz and 100Hz, at a rotational amplitude of approximately 0.5°. The test setup is shown in Figure 4.3. An oven surrounds the sample and allows the ambient temperature to be controlled.

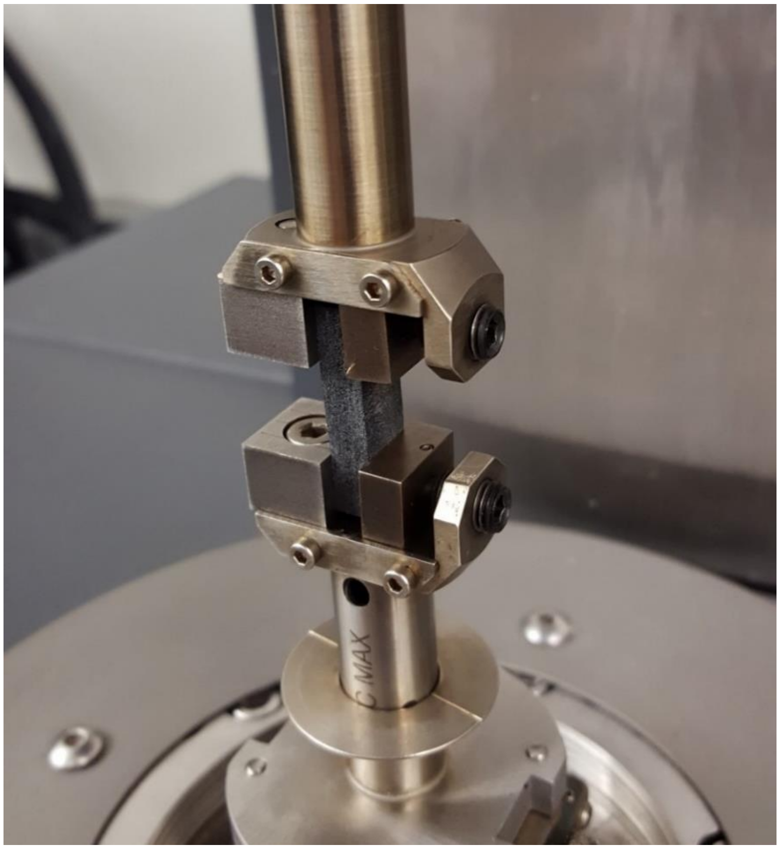


Figure 4.3. Test setup of the rubber belt sample. The lower support oscillates along the vertical axis at an amplitude of approximately 0.5° and the upper support measures the resulting torque in the material. Not shown is an oven surrounding the sample to control the object temperature.

Viscoelastic materials are characterized by two dynamic moduli to treat the elastic and viscous properties separately. The Ares 2 DMA used in this experiment is a strain controlled device and measures the amplitude and phase of the resulting torsion. From these values, the measured torsion is broken down into an in phase component, and a 90° out of phase component. The storage modulus, denoted G' for shear and E' for tension, represents the elastic response of the material from the in-phase component. For a purely elastic material, this is equivalent to the shear modulus. The loss modulus, denoted G" for shear and E" for tension, represents the viscous response from the out-of-phase component [10]. The equations for the dynamic moduli are as follows:

$$G' = \frac{\tau_o}{\gamma_o}\cos(\delta) \tag{4.1}$$

$$G" = \frac{\tau_o}{\gamma_o}\sin(\delta) \tag{4.2}$$

where τ_o and γ_o are the amplitudes of the oscillating shear stress and strain and δ is the phase shift between stress and strain [14].

In the test software for the DMA, the sample cross section dimensions are entered into the experiment settings and the distance between the sample supports is determined by the machine. From this information, it is able to calculate the values of these moduli automatically. Gates recommends that the operating temperature of a rubber belt should not exceed 160°F (71°C) for extended periods of time to extend longevity [15]. To represent the range of possible temperatures experienced by the belt in operation, four tests are conducted at ambient temperatures of 25°C, 40°C, 60°C, and 80°C. Each test sweeps the oscillation frequency from 10Hz to 100Hz. Plots of the storage and loss moduli vs frequency are seen in Figures 4.4 and 4.5. Figure 4.6 shows the relationship between both dynamic moduli and temperature with the properties evaluated at 100Hz.

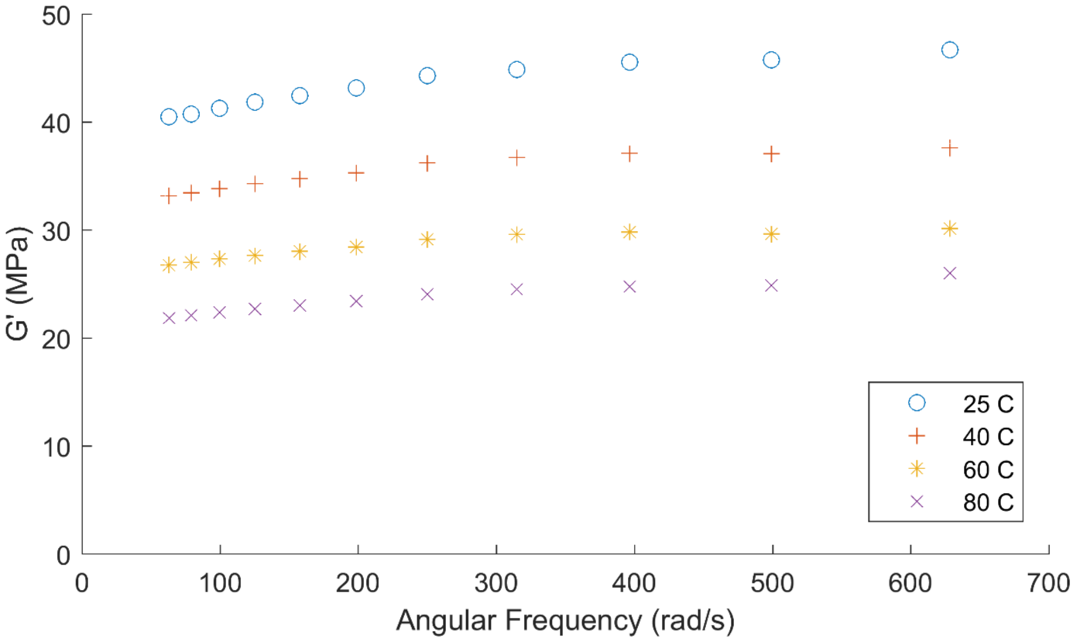


Figure 4.4. Plot of storage modulus (G’) vs frequency at various temperatures. Frequencies tested range from 10Hz to 100Hz. Storage modulus does not show a strong dependence on frequency.

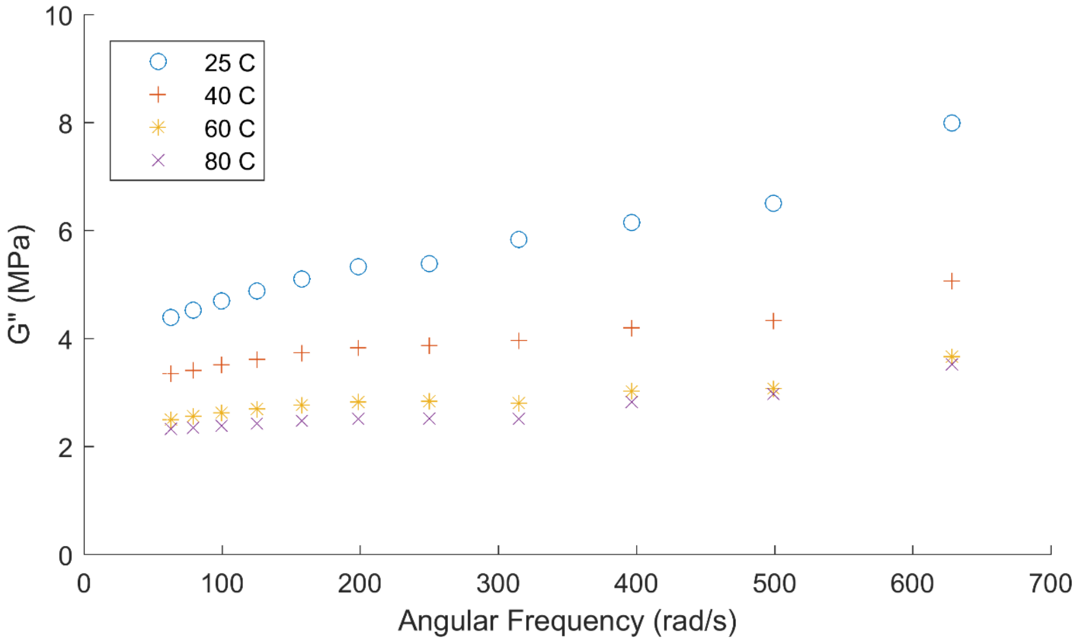


Figure 4.5. Relationship between the loss modulus (G") and frequency at various temperatures. Frequencies tested range from 10Hz to 100Hz.

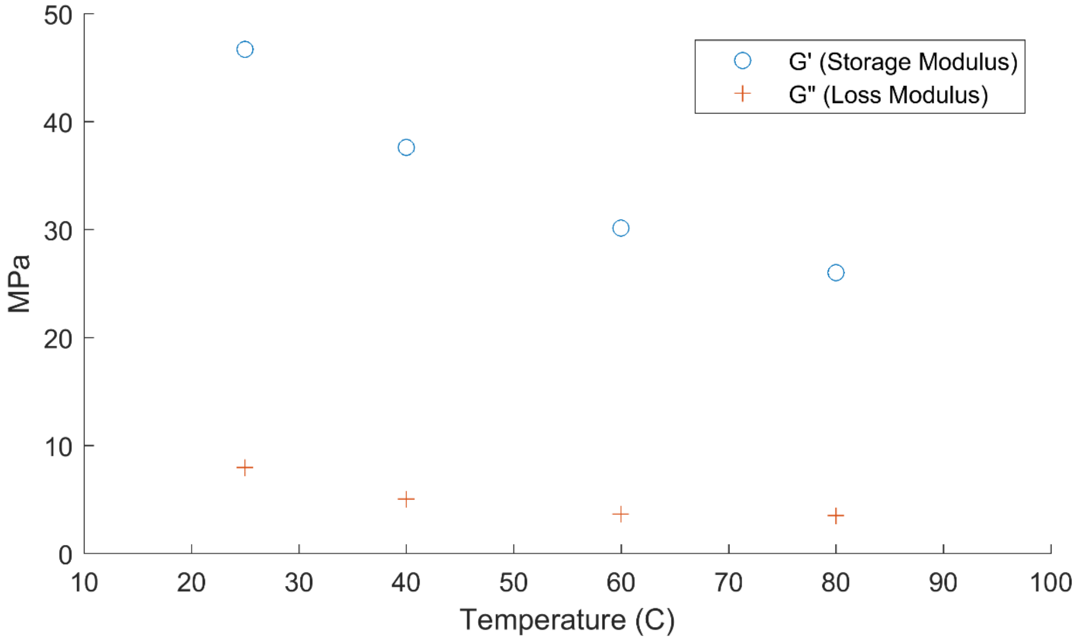


Figure 4.6. Relationship between the dynamic moduli and temperature. Both moduli decrease significantly with temperature but begin to level off around the maximum recommended operating temperature. All results are obtained at an excitation frequency of 100Hz.

The data from the DMA indicates that the elastic effects dominate over the viscous effects for all temperatures and frequencies tested. At 60°C, the magnitude of the storage modulus is 8.2 times that of the loss modulus. This ratio, along with the fact that the belt is reinforced, provides justification for the approximation that the belt behaves as a purely elastic material, and that the storage modulus is a good approximation for the shear modulus of the belt. At room temperature and 60°C, the values of the shear modulus are 46.71MPa and 30.15MPa, respectively. For an isotropic material, the shear modulus and Young's modulus are related through the following equation:

$$E = 2G(1 + \upsilon) \tag{4.3}$$

where υ is the Poison ratio of the material. For rubber, this value is approximately 0.5 and simplifies the relation:

$$E = 3G \tag{4.4}$$

From this, the Young's Moduli for the rubber at room temperature and 60°C are 140.1MPa and 90.45MPa, respectively, and these values are used for the remainder of this analysis. 60°C is well below the recommended operating temperature of 71°C, and is an estimate for the belt in a typical application. A more comprehensive study of belt temperature is beyond the scope of this thesis and more testing is required to determine this value with higher accuracy.

4.2 Aramid Fiber Reinforcement Properties

The CVTech belt is reinforced by 27 individual aramid yarns embedded in the belt cross section. The cross section of the belt is shown in Figure 4.7. These fibers are adhered to the inside of the belt and cannot be removed easily. For simplicity, these fibers are assumed to follow the properties of Kevlar 29 yarns. From the Kevlar 29 datasheet, the tensile modulus is given in units of grams-force per denier. A denier is a measure of mass per length with one denier being equal to 1g/9000m. Kevlar 29 yarns have a mass per length of 1500 denier and a tensile modulus of 555g/d [16]. Hooke's law is stated as

$$F = EA\varepsilon \tag{4.5}$$

where EA is the stiffness term. From the given quantities, EA is determined to be 8166N for a single yarn, and 220.5kN for the set of 27 yarns.

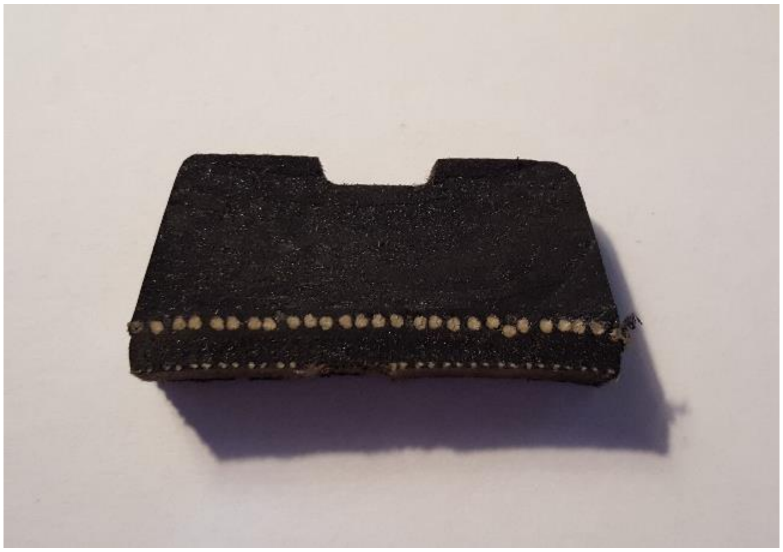


Figure 4.7. Cross section of the CVTech belt used in this analysis. The belt contains 27 aramid yarns spaced evenly in a line.

The datasheet also indicates that Kevlar does not show as strong of a dependence of material properties on temperature as rubber. The datasheet recommends that temperature effects be taken into consideration when the operating temperature exceeds approximately 150°C for extended periods of time. Since the belt is assumed to have an operating temperature of 60°C for this model, these effects are not considered.

4.3 Longitudinal Stiffness

The governing equations derived in Section 3 require the longitudinal belt effective stiffness, expressed as EA. The Young's modulus is approximated using data obtained in Section 4.1, but the cross section of the belt is not known. Since the belt has a single cog design, the cross sectional area is not constant and stress concentrations are present, causing the tensile stress to vary through the cross section. The belt tooth and width dimensions are measured with calipers and a drawing of the belt cross section is shown in Figure 4.8. The sheave angle for a CVTech pulley is known to be 12° and it is assumed that the belt wedge half-angle is equal to this.

The effective stiffness of the rubber belt is determined through the use of FEA in SolidWorks, and the results are shown in Figure 4.9. The belt model used for FEA includes 15 teeth and is 152.66mm long when unloaded. The Young's Modulus and shear modulus are entered manually into the material properties and a load of 100N is applied. The belt length increases by 0.6651mm, resulting in a strain of $4.357*10^{-3}$. Using equation 4.5, the rubber longitudinal stiffness at 60°C is 22.95kN. This stiffness is added to the yarn stiffness to obtain the total belt longitudinal stiffness of 243.5kN.

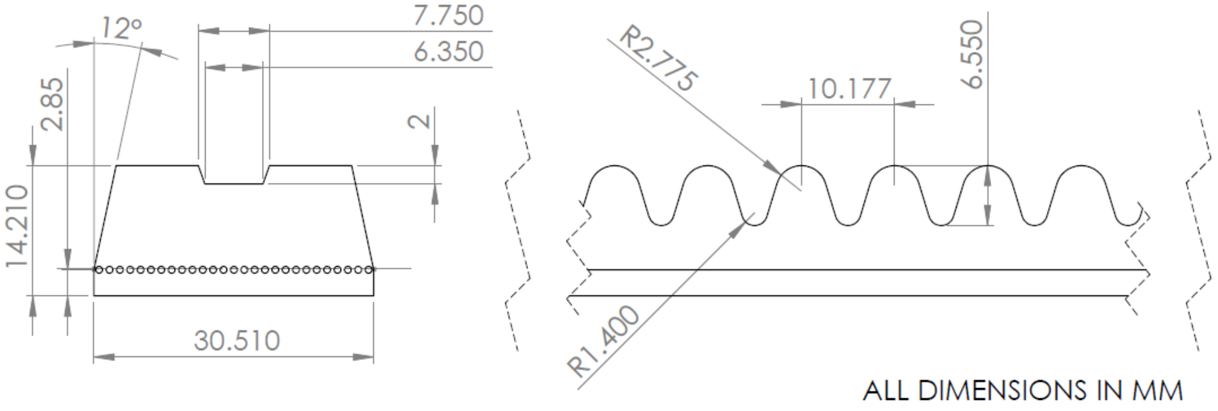
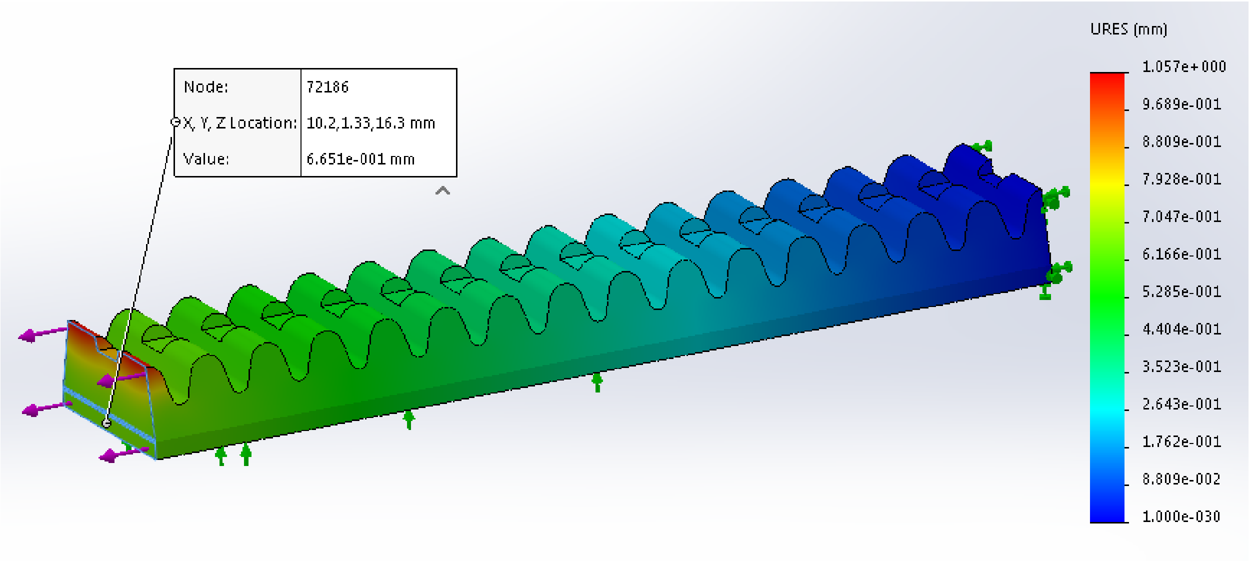


Figure 4.8. Drawing of a CVTech belt.



Node:	72186
X, Y, Z Location:	10.2,1.33,16.3 mm
Value:	6.651e-001 mm

Figure 4.9. FEA displacement results to determine the belt longitudinal stiffness. The right face of the belt is fixed and the bottom surface has a bearing constraint to prevent vertical travel and enforce displacements in the longitudinal direction only. A load of 100N applied to the left face resulted in a change in length of 0.6651mm.

4.4 Bending Stiffness

To determine the bending stiffness of the belt, the second area moment of the cross section must be determined. As before for longitudinal stiffness, the belt cross section is not constant and stress concentrations are present. An equivalent belt cross section must be determined through FEA. By cantilevering the belt and applying a known moment to the free end, measuring the approximate radius of curvature allows the section second area moment to be calculated. The same belt model used for longitudinal stiffness is used again in this section. A moment of 0.05N*m is applied to the free end and the displacement is measured. The stress concentrations due to the non-uniform cross section are seen in Figure 4.10 and the deformed result is seen in Figure 4.11. A sketch is overlaid on the deformed result and the approximate radius of curvature is measured to be 3653mm.

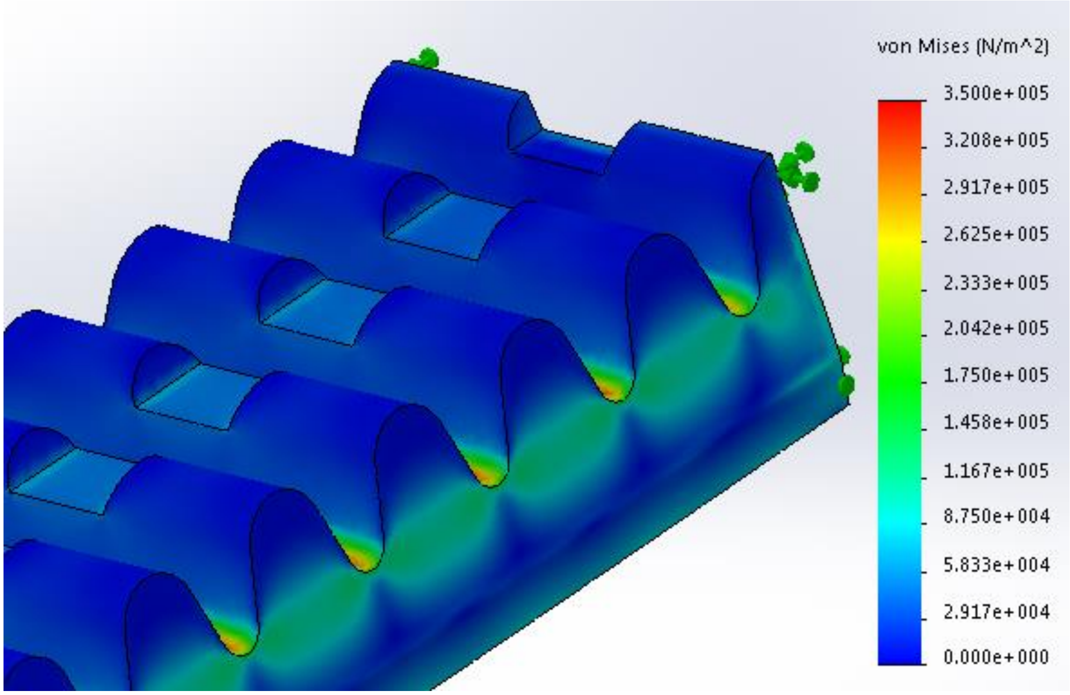


Figure 4.10. FEA results showing stress concentrations present in the CVTech belt under bending.

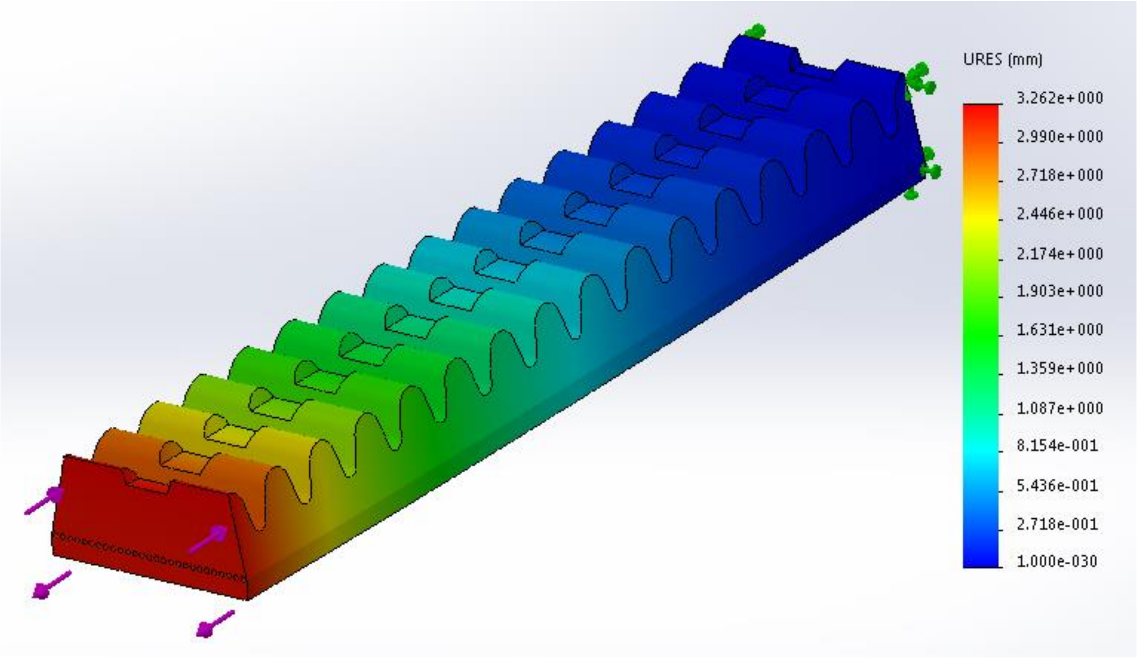


Figure 4.11. FEA displacement results from the belt bending simulation. The right face is fixed and a moment is applied to the left face. A maximum displacement of 3.262mm is observed, corresponding to a belt outside radius of 3653mm.

The belt thickness is much smaller than the length, justifying the use of Euler-Bernoulli beam theory to determine the cross section second area moment about the neutral axis through the following equation:

$$I = \frac{M\rho}{E} \tag{4.6}$$

The radius measured from FEA corresponds to the belt's outside radius and serves as an approximation for ρ since the belt thickness is much smaller than the radius of curvature. From this, the second area moment about the neutral axis is 2.019*$10^{-9}$$m^4$. The longitudinal stiffness of the aramid yarns is much greater than that of the rubber and has the effect of moving the neutral surface to the location of the yarns. To be able to use the parallel axis theorem to determine the resulting second area moment, the original location of the neutral surface needs to be determined. As seen in Figure 4.8, the belt cross section consists of a rectangle below the yarns and a trapezoid above the yarns. All dimensions are known except the trapezoid height. A system of algebraic equations are solved in MATLAB to determine the height that results in the second area moment as previously determined. Code for these equations is included in the Appendix, and the results are illustrated in Figure 4.12. The results indicate that the equivalent cross section is 6.532mm tall and the neutral axis is 1.759mm above the yarns. The parallel axis theorem is

$$I = \bar{I} + Ad^2 \tag{4.7}$$

where $\bar{I}$ is the second area moment about the neutral axis, A is the area of the cross section, and d is the distance between the axes. From the results in MATLAB, the area is 2.772*10^{-4}m^2, resulting in a new second area moment of 2.877*10^{-9}m^4. Multiplying this by Young's modulus at 60°C gives a bending stiffness 0.2602N*m^2.

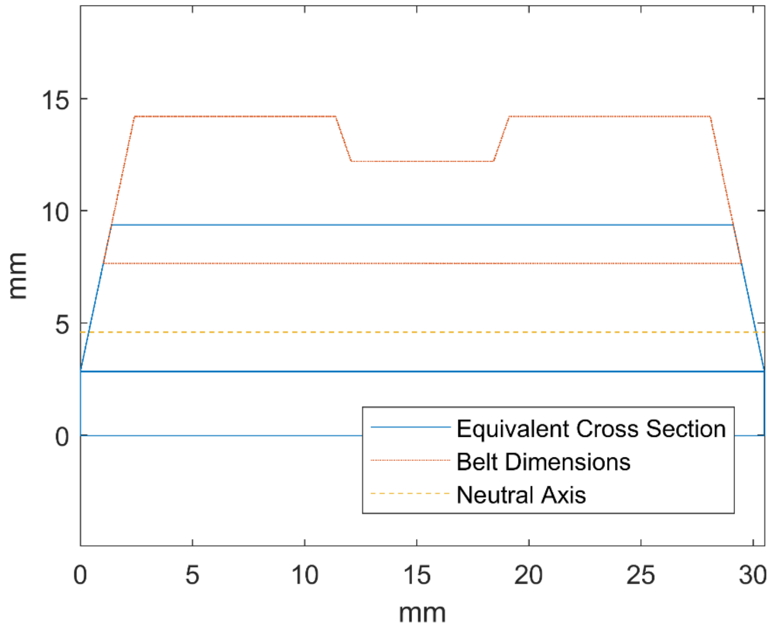


Figure 4.12. Belt equivalent cross section in bending as determined in MATLAB. The belt dimensions indicate the top and bottom of the tooth pattern, and the equivalent cross section falls between these limits.

4.5 Lateral Stiffness

The belt lateral stiffness is measured directly using an INSTRON load cell. Two aluminum blocks each cut at an angle of 12° represent the contact with the sheaves, and force the belt to retain the same geometry as it is compressed. This apparatus is shown in Figure 4.14 and the length of the aluminum blocks and belt sample are 51.25mm. The large stiffness of the aluminum blocks compared to the rubber allows them to be treated as rigid and their inclusion does not affect the accuracy of the data. The INSTRON is displacement controlled and the sample was loaded at a constant travel rate of 4mm/min, with the maximum load set to 1200N. The data are shown in Figure 4.15. A linear fit with a slope of 1747N/mm matches the data well with an R^2 value of 0.9918.

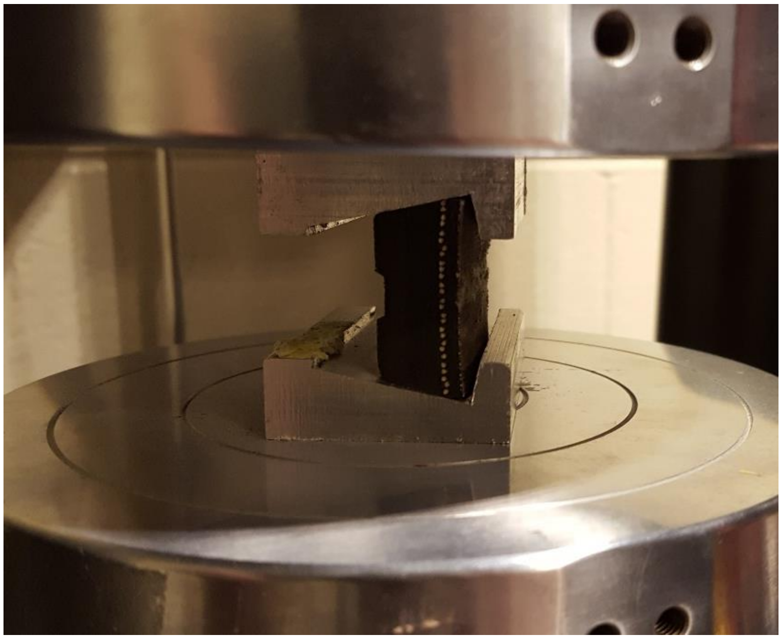


Figure 4.13. Test setup for measuring the belt lateral stiffness. Two aluminum blocks represent the sheave angle.

The belt model requires the belt stiffness to be expressed as a radial spring stiffness, k. This quantity is defined as a force per length on the belt per radial displacement. The sheave angle, β relates the radial displacement, x, to the lateral displacement, D_L, through the following equation:

$$x = \frac{D_L}{2\tan(\beta)} \qquad (4.8)$$

Equation 3.16 relates the radial stiffness to the radial displacement, and axial force per length, and is repeated here:

$$k = \frac{2p_z}{x} \tag{4.9}$$

The axial force is related to the measured force, F, and the sample length, L_S, as follows:

$$p_z = \frac{F}{L_S} \tag{4.10}$$

This yields the following expression for k with the measured slope, m, expressed in N/m:

$$k = 2\frac{F}{L_S}\frac{2\tan(\beta)}{D_L} = 4\frac{m}{L_S}\tan(\beta) \tag{4.11}$$

Substituting in known values yields 28.98MPa for k at room temperature. The units indicate this is a force per length of the belt, per length radial penetration. The load cell used for this test does not have the capability of changing the ambient temperature of the sample and cannot obtain data for any other temperatures. To estimate the belt radial spring stiffness at 60°C, it is assumed that the stiffness is directly proportional to the elastic modulus and multiplying by the ratio of the moduli yields a value for k at 60°C of 18.71MPa.

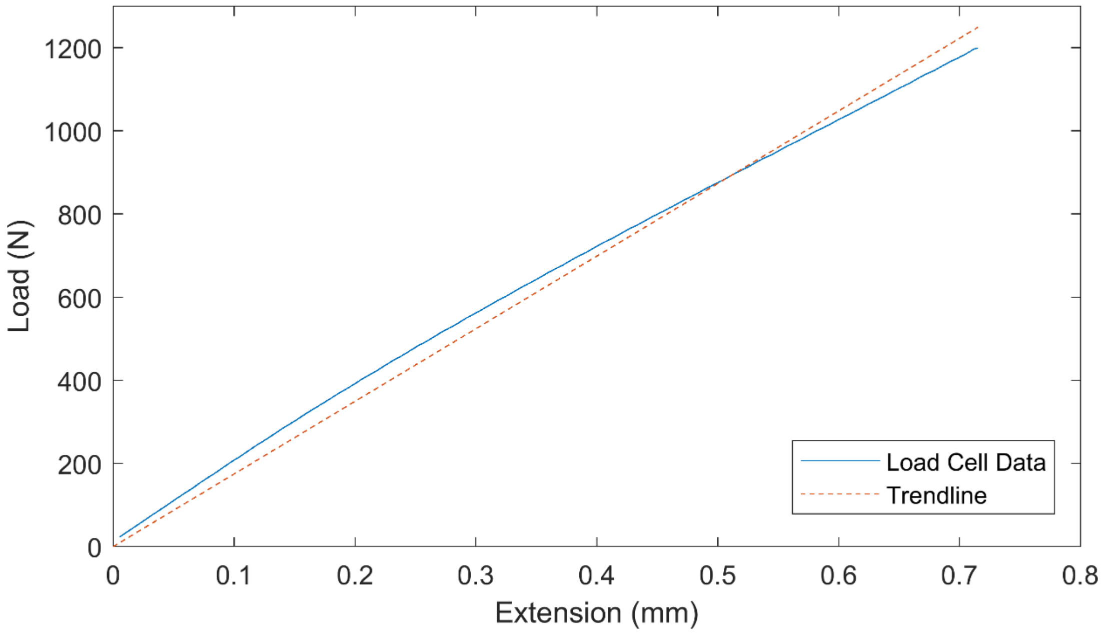


Figure 4.14. Load cell data for the belt lateral stiffness. The data follows a linear trend in this region with a calculated slope of 1747N/mm and an R^2 value of 0.9918.

4.6 Pitch length

The outside circumference of the belt is measured to be 875.1mm. For this model, it is assumed that the neutral surface is at the location of the fiber reinforcements. The reinforcements are located 2.85mm from the outside surface of the belt. Treating the belt as a circle in its unloaded condition, the pitch length is then calculated to be 857.2mm, which is used in this model. To determine the mass per length of the belt, m_o, the total mass of the belt is measured to be 375g. Dividing by the pitch length gives a linear mass density of 0.437 kg/m. From the datasheet, this belt is intended to be used with a pulley center-to-center distance of 214mm [2].

5 Numerical Methods

The equations derived in Section 3 constitute a boundary value problem. These equations require corresponding boundary conditions imposed at the beginning and end of the region of interest to obtain a solution. Two methods commonly used to solve BVP's are the shooting method and collocation. Due to the complexity of the problem and the distribution of boundary conditions, collocation is used, and bvp4c, the built-in BVP solver in MATLAB, implements this method. Since the length of the solution region is not known initially due to the unknown wrap angle and span lengths, the equations must be converted into a non-dimensional form to obtain a solution. The boundary conditions are then added to the system and a solution is obtained through continuation by simplifying the equations and modifying the parameters to obtain a solution, and gradually making adjustments to the actual equations and values, which will be explained later.

5.1 Solution Approaches for Boundary Value Problems

First-order systems of ordinary differential equations may be posed as either boundary value or initial value problems. For both types, the number of conditions provided must be equal to the number of first-order equations. Initial value problems are typically solved using explicit Runge-Kutta methods. These methods involve using already calculated values to determine the next solution step and ode45 is one example of a built-in function in MATLAB that uses this method [17]. Stiff ODE's are defined as equations where the step size required for convergence would be unreasonably small and other built-in solvers, such as ode15s, are optimized for working with these kinds of equations [18].

Two common methods for solving boundary value problems are the shooting method and collocation. The shooting method involved treating the system as if it was an initial value problem. Conditions applied at the right end are removed and an equal number of conditions are added to the left end. These stand-in conditions are then adjusted until the original conditions at the right end are satisfied. This method takes advantage of the efficiency and robustness of existing methods to solve initial value problems, and works well when there is only one condition on the right side or the solution is sensitive to a change in initial conditions [17]. Gerbert and Sorge use this method in [12] to solve for the belt path for a single pulley. In order to obtain a solution, assumptions are made about the behavior of the belt path to limit the possible combinations of the stand-in initial conditions, and the actual conditions are determined through iteration.

Another option for solving boundary value problems is through collocation, and bvp4c is a built-in function in MATLAB that uses this method. In addition to the equations and boundary conditions, this function requires the user to specify guesses for the solution at mesh points. The solver then refines the mesh as necessary and iterates to arrive at a solution that satisfies the boundary conditions. This function is used by Kong and Parker in [6] to solve the governing equations for both pulleys simultaneously in a V-belt string model. Due to the number of conditions imposed at the right end and the insensitivity of the equations to the boundary conditions, bvp4c is used to solve the equations derived in Section 3.

5.2 Conversion to Non-Dimensional Form

For a two point boundary value problem, all conditions must be specified at either the left or right end of the solution region. The governing equations consist of four solution regions; two contact regions, and two belt spans. For the contact regions, the wrap angle is not known initially, and for the spans the arc length is not known either. This results in a boundary value problem with a solution region of unknown size, and this is not compatible with the standard form required by bvp4c. Ascher and Russel propose a method to solve for unknown constants in a boundary value problem by including the value as a solution variable, with its derivative set to zero [19]. Kong and Parker extend this to the solution method in [6]. The equations are non-dimensionalized such the independent variables are divided by the overall length of the solution region. This results in independent variables on the range of 0 to 1, and the scaling factor is treated as an unknown constant to determine the wrap angles and arc lengths. The non-dimensional wrap angle for the contact regions, $\hat{\phi}$, is defined in terms of the total wrap angle, Φ, as follows:

$$\phi = \Phi\,\hat{\phi} \tag{5.1}$$

Similarly, relationships between derivatives with respect to the dimensional and non-dimensional variables can be shown:

$$\frac{d}{d\phi} \rightarrow \frac{1}{\Phi}\frac{d}{d\hat{\phi}} \tag{5.2}$$

$$\frac{d^2}{d\phi^2} \rightarrow \frac{1}{\Phi^2}\frac{d^2}{d\hat{\phi}^2} \tag{5.3}$$

$$\frac{d^3}{d\phi^3} \rightarrow \frac{1}{\Phi^3}\frac{d^3}{d\hat{\phi}^3} \tag{5.4}$$

These expressions are plugged into the governing equations to result in the following non-dimensional forms, where a hat denotes a variable that is derived with respect to $\hat{\phi}$:

$$\hat{T}' = \Phi\left[2p\;(-\sin\beta\sin\theta + \mu\cos\beta_s\sin(\theta+\gamma))\;\frac{R-x}{\cos\theta} + \left(\frac{\hat{\theta}'}{\Phi} - 1\right)\left(-\frac{EI}{R^2}\frac{\hat{\theta}''}{\Phi^2} + \frac{EI}{R^3}\frac{\hat{x}'}{\Phi}\right)\right] \tag{5.5}$$

$$\hat{\theta}''' = \frac{TR^3\Phi^3}{EI(R+2\,x)}\left(1 - \frac{2p}{T}\;(\sin\beta\cos\theta - \;\mu\cos\beta_s\cos(\theta+\gamma))\;\frac{R-x}{\cos\theta} - \frac{\hat{\theta}'}{\Phi} + \frac{1}{T}\frac{EI}{R^3}\frac{\hat{x}''}{\Phi^2}\right) \tag{5.6}$$

$$x'' = -\Phi^2\left[\frac{(R-x)}{\cos(\theta)^2}\frac{\hat{\theta}'}{\Phi} + \tan(\theta)\,\frac{\hat{x}'}{\Phi}\right] \tag{5.7}$$

An equation to solve for the unknown wrap angle is also introduced:

$$\hat{\Phi}' = 0 \tag{5.8}$$

In the span regions, a total arc length, L, is specified and the following conversions are used to define the non-dimensional arc length, $\hat{s}$:

$$s = \mathrm{L}\,\hat{s} \tag{5.9}$$

$$\frac{d}{ds} \rightarrow \frac{1}{\mathrm{L}}\frac{d}{d\hat{s}} \tag{5.10}$$

$$\frac{d^2}{ds^2} \rightarrow \frac{1}{L^2}\frac{d^2}{d\hat{s}^2} \tag{5.11}$$

$$\frac{d^3}{ds^3} \rightarrow \frac{1}{L^3}\frac{d^3}{d\hat{s}^3} \tag{5.12}$$

Plugging these expressions into the span governing equations yield the following equations where a hat denotes a derivative with respect to $\hat{s}$:

$$\hat{\tau}' = -\frac{EI}{L^2}\,\hat{\vartheta}'\hat{\vartheta}'' \tag{5.13}$$

$$\hat{\vartheta}''' = L^2\frac{\tau\hat{\vartheta}'}{EI} \tag{5.14}$$

Similarly, an equation for the unknown arc length is also introduced:

$$\hat{\mathrm{L}}' = 0 \tag{5.15}$$

Another unknown constant in the system is ω_S, the secondary pulley angular velocity. This constant is solved for in the same manner as the arc lengths and wrap angles, with the following equation also added to the system:

$$\omega_{\mathrm{S}}' = 0 \tag{5.16}$$

5.3 Boundary Conditions

To be able to solve the non-dimensionalized equations, a number of boundary conditions equaling the total number of first-order equations is needed. To obtain these conditions, certain parameters related to the geometry of the system must be defined. Some boundary conditions are included to force continuity in the belt path between contact and span regions, while others are user-defined to specify the operating point of the belt drive.

5.3.1 Geometry Definitions

In the string model presented in [6], boundary conditions are used to account for the belt trajectory in the spans. When bending stiffness is not included, the span trajectory will be straight, forcing the exit and entry angles for both pulleys for a given span to match. In [6], a coordinate system is defined relative to the slack span and all other angles are defined relative to this. When bending stiffness is included, the spans will no longer be straight and another coordinate system must be used. Two angles, C_P, and C_S, are defined to determine the orientation of the pulleys. For the primary pulley, this angle is defined as between the entry point, and the line between the pulley centers. For the secondary pulley, this angle is between the exit point and the line between the pulley centers. This is illustrated in Figure 5.1, as well as the location and orientation of an x-y coordinate system. Combined with the total wrap angle, Φ, these angles are used to determine the location of the entry and exit points of the belt. The primary pulley entry and exit points are denoted as P_{Pt}, and P_{Ps}, respectively, and the secondary entry and exit points are denoted as P_{Ss}, and P_{St}, respectively. The x and y coordinates of these points are determined though the following relations:

$$P_{Pt}(x,y) = (R_P \cos(C_P)\,, R_P \sin(C_P)) \tag{5.17}$$

$$P_{Ps}(x,y) = (R_P \cos(C_P + \Phi_P)\,, R_P \sin(C_P + \Phi_S)) \tag{5.18}$$

$$P_{Ss}(x,y) = (L_c - R_S \cos(C_S)\,, R_S \sin(C_P)) \tag{5.19}$$

$$P_{St}(x,y) = (L_c - R_S \cos(C_S + \Phi_S)\,, R_S \sin(C_S + \Phi_S)) \tag{5.20}$$

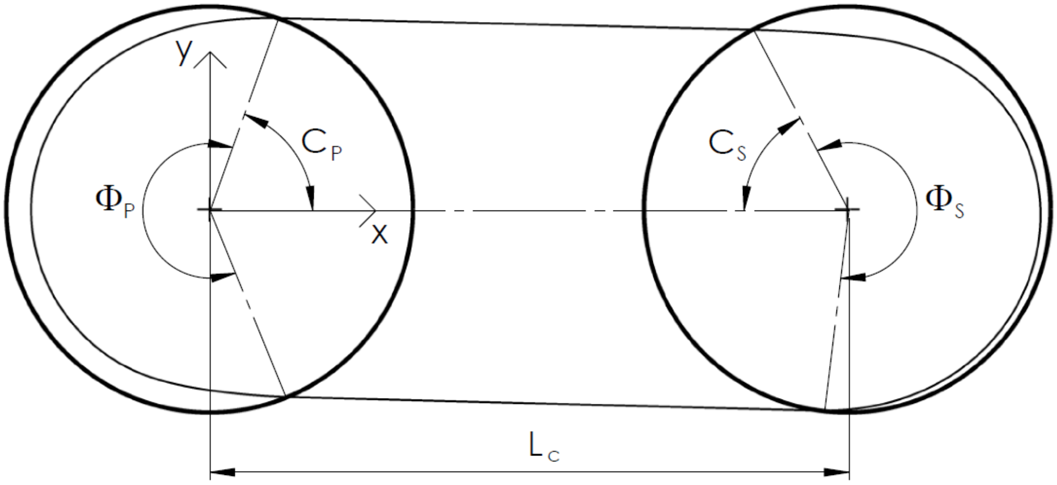


Figure 5.1. Definition of the pulley orientation angles. These angles are used to define the system geometry and enforce continuity in the belt path between the contact and span regions. The location and orientation of the x-y coordinate system is also shown, as well as the pulleys' center-to-center distance.

Entry and exit angles of the belt are also determined using the pulley orientation angles. These angles are relative to the positive x-axis and are defined as follows:

$$E_{Pt} = C_P - \frac{\pi}{2} - \theta_{P1} \tag{5.21}$$

$$E_{Ps} = C_P + \Phi_P - \frac{3\pi}{2} - \theta_{P2} \tag{5.22}$$

$$E_{Ss} = \frac{3\pi}{2} - C_S - \Phi_S + \theta_{S1} \tag{5.23}$$

$$E_{St} = \frac{\pi}{2} - C_S - \theta_{S2} \tag{5.24}$$

where θ is the belt angle of inclination, and subscripts 1 and 2 denote the beginning and end of the solution region, respectively. Both pulleys pictured are assumed to be rotating counter-clockwise, and all solution regions are solved in the direction of the belt travel. This implies that the tight and slack spans correspond to the upper and lower spans, respectively.

In the span regions, x and y coordinates of the belt are determined through the angle of inclination. The x and y solution variables are non-dimensionalized and their derivatives with respect to arc length are as follows:

$$x = L\,\hat{x} \qquad \hat{x}' = \cos(\theta) \tag{5.25}$$

$$y = L\,\hat{y} \qquad \hat{y}' = \sin(\theta) \tag{5.26}$$

5.3.2 Continuity Boundary Conditions

From the given geometry definitions, boundary conditions are imposed to enforce continuity in the belt path between contact and span regions. The x and y coordinates of the exit points, as determined in Equations 5.17 through 5.20 must match the corresponding values from the span regions. These boundary conditions are shown in Equations 5.27 through 5.34. Subscripts t and s denote the tight and slack spans, respectively.

$$R_P \cos(C_P) - L_t\, \hat{x}_{t2} = 0 \tag{5.27}$$

$$R_P \sin(C_P) - L_t\, \hat{y}_{t2} = 0 \tag{5.28}$$

$$R_P \cos(C_P + \Phi_P) - L_s\, \hat{x}_{s1} = 0 \tag{5.29}$$

$$R_P \sin(C_P + \Phi_S) - L_s\, \hat{y}_{s1} = 0 \tag{5.30}$$

$$L_c - R_S \cos(C_S + \Phi_S) - L_s\, \hat{x}_{s2} = 0 \tag{5.31}$$

$$R_S \sin(C_S + \Phi_S) - L_s\, \hat{y}_{s2} = 0 \tag{5.32}$$

$$L_c - R_S \cos(C_S) - L_t\, \hat{x}_{t1} = 0 \tag{5.33}$$

$$R_S \sin(C_P) - L_t\, \hat{y}_{t1} = 0 \tag{5.34}$$

Similarly, the exit angles calculated in Equations 5.21 through 5.24 must match the corresponding angles of inclination from the span region. These boundary conditions are shown in Equations 5.35 through 5.38. Since the angles of inclination for the tight span are relative to the negative x direction, the exit and entrance angles need to be rotated by π to match them.

$$E_{Pt} + \pi - \vartheta_{t2} = 0 \tag{5.35}$$

$$E_{Ps} - \vartheta_{s1} = 0 \tag{5.36}$$

$$E_{Ss} - \vartheta_{s2} = 0 \tag{5.37}$$

$$E_{St} + \pi - \vartheta_{t1} = 0 \tag{5.38}$$

Equation 3.27 relates the transverse shear to the belt internal moment, which is inversely proportional to the curvature by Euler-Bernoulli beam theory. Any discontinuity in the radius of curvature would cause a step change in the internal moment, and this would result in an infinite shear force. This implies that the radius of curvature must be continuous throughout the entire belt path. The boundary conditions used to enforce this continuity are shown in Equations 5.39 through 5.42. Equations 3.2 through 3.4 are used to convert the derivative with respect to wrap angle in the contact regions into a derivative with respect to arc length, corresponding to the curvature.

$$\frac{\cos(\theta_{P1})}{R_P - x_{P1}}\left(1 - \frac{\hat{\theta}'_{P1}}{\Phi_P}\right) - \frac{\hat{\vartheta}'_{t2}}{L_t} = 0 \tag{5.39}$$

$$\frac{\cos(\theta_{P2})}{R_P - x_{P2}}\left(1 - \frac{\hat{\theta}'_{P2}}{\Phi_P}\right) - \frac{\hat{\vartheta}'_{s1}}{L_s} = 0 \tag{5.40}$$

$$\frac{\cos(\theta_{S1})}{R_S - x_{S1}}\left(1 - \frac{\hat{\theta}'_{S1}}{\Phi_S}\right) - \frac{\hat{\vartheta}'_{s2}}{L_s} = 0 \tag{5.41}$$

$$\frac{\cos(\theta_{S2})}{R_S - x_{S2}}\left(1 - \frac{\hat{\theta}'_{S2}}{\Phi_S}\right) - \frac{\hat{\vartheta}'_{t1}}{L_t} = 0 \tag{5.42}$$

Boundary conditions are also used to constrain the transverse shear at the interfaces. In [7], Gerbert attempts to calculate the shear force for a V-belt at the entrance and exit points on the pulley by drawing comparisons with a flat belt that is radially compressible. In this model, it is assumed that all tension is carried by the reinforcing material in the belt, and this material is located on the outside surface of the belt. This model allows the tension to act in a direction not parallel to the reinforcing material. This would seem to imply that the transverse shear in the belt at the entry and exit points would be zero when there is no radial compression. In [11], the flat belt that is analyzed is assumed to be rigid in the radial direction, forcing the radius of curvature to be constant until the belt leaves the pulley, implying that the transverse shear is zero at the departure point on the pulley side. The V-belt model presented in this thesis assumes the belt follows line contact along the pulley, with no radial compression present in the belt. For this reason, the boundary conditions are chosen to enforce zero transverse shear in the belt at the entry and exit points on the pulley side. The linear approximation for the transverse shear in the belt is given in Equation 3.38. This term is set to zero at the ends of the contact region to yield the following boundary conditions:

$$-\hat{\theta}''_{P1} + \frac{\hat{x}'_{P1}}{R_P} = 0 \tag{5.43}$$

$$-\hat{\theta}''_{P2} + \frac{\hat{x}'_{P2}}{R_P} = 0 \tag{5.44}$$

$$-\hat{\theta}_{S1}'' + \frac{\hat{x}_{S1}'}{R_S} = 0 \tag{5.45}$$

$$-\hat{\theta}_{S2}'' + \frac{\hat{x}_{S2}'}{R_S} = 0 \tag{5.46}$$

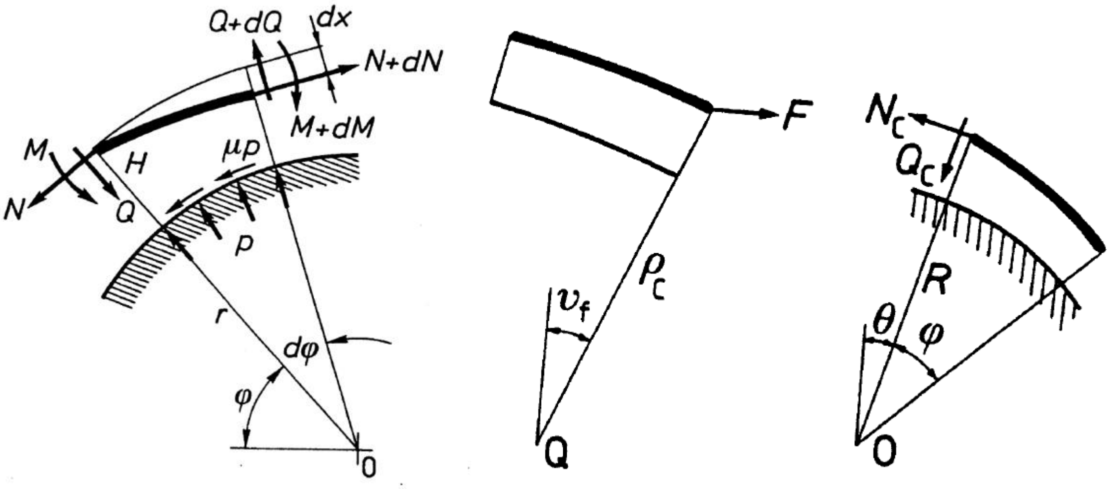


Figure 5.2. Flat belt radial flexure model [7]. Pictured left are the forces present on a belt differential element and pictured right are two belt differential elements at the departure point, before and after exiting the pulley.

Tension must also be continuous at the interfaces since no external forces are present. This leads to the following boundary conditions:

$$T_{P1} - \tau_{t2} = 0 \tag{5.47}$$

$$T_{P2} - \tau_{s1} = 0 \tag{5.48}$$

$$T_{S1} - \tau_{s2} = 0 \tag{5.49}$$

$$T_{S2} - \tau_{t1} = 0 \tag{5.50}$$

5.3.3 Radial Penetration Boundary Conditions

By definition, the radial penetration at the ends of the contact regions must be zero as the belt enters or exits the pulley:

$$x_{P1} = x_{P2} = x_{S1} = x_{S2} = 0 \tag{5.51}$$

Equation 3.42 is obtained by deriving Equation 3.24. This is necessary to provide an expression for x'' in Equation 3.41. However, this creates an additional equation when converted to first order form and requires an additional boundary condition to be added to the system. When derived, any constants from Equation 3.24 would be lost. By definition, the expression for x' must still be true everywhere in the contact region, and may be used as a boundary condition at an arbitrary point to add any integration constants back into the system. This arbitrary point is chosen to be the entry point for both pulleys:

$$(R_P - x_{P1}) * \tan(\theta_{P1}) + \frac{\hat{x}'_{P1}}{\Phi_P} = 0 \tag{5.52}$$

$$(R_S - x_{S1}) * \tan(\theta_{S1}) + \frac{\hat{x}'_{S1}}{\Phi_S} = 0 \tag{5.53}$$

5.3.4 Integrated Quantities

Several additional quantities are included in the system of equations that are integrals across the solution region. While it would be possible to determine these values by numerically integrating the solution vector, it is easier to include these equations with the rest of the system. A benefit to this approach is that the value of the integral may be used as a boundary condition. One of the integrated quantities is the pulley torque. The equation for this is given in [6] and converted to non-dimensional form here:

$$\widehat{M}' = \Phi \frac{2\, \mu\, p \cos(\beta_s) \sin(\gamma)\ \ (R - x)^2}{\cos(\theta)} \tag{5.54}$$

The belt unstretched length is integrated over the solution regions from the tension and the belt longitudinal stiffness. This value is iterated for in [11] to determine the tension in a flat belt system. For a V-belt, this value can be compared to the measured belt value to evaluate the accuracy of the model.

The non-dimensional derivatives of unstretched length with respect to wrap angle and arc length are given in Equations 5.55 and 5.56, respectively.

$$\hat{\mathcal{L}}_o' = \Phi\frac{R-x}{\cos(\theta)}\frac{1}{1+\frac{F}{EA}} \tag{5.55}$$

$$\hat{\mathcal{L}}_o' = \frac{L}{1+\frac{F}{EA}} \tag{5.56}$$

Another integrated quantity is the axial force in the pulley. This value is calculated from the radial penetration and the belt radial spring stiffness. Equation 3.16 relates the axial component of the sheave pressure on the belt and is repeated here:

$$p_z = \frac{k\,x}{2} \tag{5.57}$$

The quantity p_z is in units of force per length and is rearranged in terms of an axial force, A, for the belt differential element. This is converted into a derivative with respect to wrap angle using Equation 3.4, then into non-dimensional form.

$$dA = p_z\,ds = \frac{k\,x}{2}ds \tag{5.58}$$

$$\frac{dA}{d\phi} = \frac{R-x}{\cos(\phi)}\frac{k\,x}{2} \tag{5.59}$$

$$\hat{A}' = \phi\frac{R-x}{\cos(\phi)}\frac{k\,x}{2} \tag{5.60}$$

The integrals of the axial force and moment at the entry point for both pulleys is zero, yielding the following boundary conditions:

$$\hat{A}_{P1} = \hat{A}_{S1} = 0 \tag{5.61}$$

$$\widehat{M}_{P1} = \widehat{M}_{S1} = 0 \tag{5.62}$$

5.3.5 Operating Point

All boundary conditions specified above are true regardless of the operating point of the system. Two additional boundary conditions are required to establish the operating point. For an ideal CVT, the engine will operate at a constant speed regardless of the secondary speed. Due to the relationship between speed and torque for an internal combustion engine (see Figure 1.1), this implies that the torque applied to the primary is ideally constant, and known from dynamometer data. This justifies using the primary torque as a boundary condition in the model. This condition is used to determine the difference in tensions between the slack and tight spans. The remaining boundary condition is used to define the slack span tension at the exit point of the primary. The value of the tractive tension in the span cannot be less than or equal to zero, and this allows the performance of the system to be analyzed as it approaches this limiting condition.

5.4 Overview of Continuation Methods

MATLAB's built in boundary value problem solver, bvp4c, requires an initial guess for the solution. Boundary value problems do not always have unique solutions, and this guess ensures that the solution obtained is the one of interest. It also aids in the convergence of the solution. For systems of equations that are linear or stable, the accuracy of the guess is not critical. For unstable, nonlinear systems, such as the one presented here, the guess must closely resemble the solution for the solver to converge. One method for obtaining solutions for unstable equations is through the use of continuation [17]. This method involves simplifying the problem to create a system of equations that are easier to solve, then gradually adjusting the equations, using the solution from each step as the initial guess for the next step until the equations reach the original forms. The formal definition for continuation is as follows [17]:

$$y' = \mu\, f(x, y) + (1 - \mu)\, F(x, y) \tag{5.63}$$

$$0 = \mu\, g\big(y(a), y(b)\big) + (1 - \mu)\, G\big(y(a), y(b)\big) \tag{5.64}$$

The original ODE function is $f(x, y)$, and the simplified function is $F(x, y)$. For the boundary conditions in Equation 5.64, $g\big(y(a), y(b)\big)$ represents the original set of boundary conditions as functions of $y(a)$ and $y(b)$, which are values of y at the left and right ends, respectively. $G\big(y(a), y(b)\big)$ represents the simplified boundary conditions. The continuation parameter, μ, is then adjusted gradually from 0 to 1. This method applies to equations, but the same concept can be applied to parameters by gradually adjusting them from a start value to an end value. The solution to the string model presented in [6] is provided by Kong. This is used as a starting point for continuation methods. The first step to adjusting this solution for the full belt bending stiffness model presented here is by adding the effects of bending stiffness into the contact regions. Adding bending stiffness to governing equations increases the order of the equations for θ and x. To provide guesses for higher derivatives of the variables, finite difference methods are used. MATLAB treats the solution to a boundary value problem as continuous and has a function, deval, that is capable of evaluating the solution at points other than the output mesh points [20]. To increase the accuracy of these finite difference approximations, the deval function is used to refine the solution mesh. The bending stiffness, EI, is initially set to a small value and the solver is allowed to converge.

Once the contact region equations are adjusted to include belt bending stiffness, the span equations must be added to the system. To provide an initial guess for the solution for the span equations, values are calculated from the solution of the previous system. The previous solution provides the locations of the entry and exit points through the pulley angle of orientation, and assumes that the spans are straight. From these given lines, values for the x and y location of the belt at mesh points are calculated. To complete the guess structure, it is assumed that the span tension and angle of inclination are constant along the span. The bending stiffness of the belt is set to a small value, and the guess is sufficiently accurate to obtain a solution. With a solution obtained for the full belt path with bending stiffness, the value of the belt bending stiffness is gradually increased until it matches the value obtained experimentally.

The bvp4c function allows the user to specify the error tolerance properties for the solver. The residuals may be defined as either absolute, or relative to the magnitude of the function. For simplicity, a relative tolerance is specified at the default value of 1*10^{-3}, implying a solution accuracy of 0.1% [21]. Smaller tolerances require finer solution meshes and increase computation time. The computation time is also significantly reduced by vectorising the equations. This is an optional property of the solver, and involves modifying the notation of the ODE function such that the solution mesh is solved as a single vector operation, instead of evaluating the equations point by point [21].

6 Model Results

Results of interest in this model are the axial forces in the pulleys and the efficiency of the system. These parameters are useful in the custom CVT design process for the Baja team by giving insight into the magnitude of the axial forces that the pulleys need to generate, as well as providing estimates of efficiency that may be used to guide design decisions. The system is solved initially for the special case where the nominal radii of the pulleys are equal, resulting in a 1:1 ratio as defined by geometry. The solution is then extended to the operating range of the CVT from low range to high range. The model may then be used to determine the effects of varying parameters on the efficiency of the transmission throughout the operating range.

6.1 Results for the 1:1 Ratio Case

Several assumptions are made to determine the radii of the pulleys. In the case of the low power application as seen in Baja SAE, belt radial penetrations are small. Belt extension in the longitudinal direction is also assumed to be small. From this, the pulley radii are approximated from the belt pitch length, L_P, and the center to center distance, L_c, as follows:

$$R_P = R_S = \frac{L_P - 2\,L_c}{2\pi} \tag{6.1}$$

The given pitch length of 857.2mm and center to center distance of 214mm result in radii of 68.3mm. A summary of the parameters used for the initial 1:1 ratio solution are shown in Table 6.1. The boundary condition for primary torque is chosen to be 18N*m. This is approximately the output torque of the engine used for Baja SAE at 3400RPM. The slack side tension is arbitrarily chosen to be 100N. Gerbert states in [7] that typical kinetic friction coefficients range from 0.3 to 0.5. The authors of [12] and [6] both choose this value to be 0.4, and this is repeated here. A plot of the belt trajectory is shown in Figure 6.1. To generate all plots of solution variables, the deval function is used to evaluate the solution at 1000 evenly spaced points to give a better representation of the continuity of the solution.

Table 6.1. Parameters used in 1:1 ratio case. Values are consistent with typical operating conditions for the CVT used with Baja SAE.

Variable	Name	Value
R_P	Primary Nominal Radius	68.3mm
R_S	Secondary Nominal Radius	68.3mm
L_c	Center-to-Center Distance	214mm
EA	Belt Longitudinal Stiffness	243.5*10^3N
EI	Belt Bending Stiffness	0.2602N/m^2
k	Belt Radial Spring Stiffness	18.71*10^6N/m^2
ω_P	Primary Angular Velocity	356.0rad/s (3400RPM)
m_o	Belt Unloaded Linear Mass Density	0.437kg/m
μ	Coefficient of Kinetic Friction	0.4
β	Sheave Angle	12°
T_s	Slack Span Tractive Tension	100N
M_P	Primary Torque	18N*m

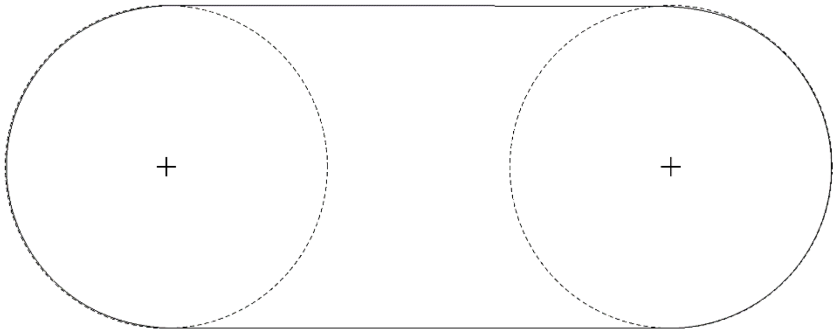


Figure 6.1. Plot of belt trajectory for the 1:1 ratio case. Nominal pulley radii are indicated by the dashed lines. Belt radial penetrations are small and span curvature is not easily distinguishable on this scale.

6.1.1 Contact Region Results

Plots of the tractive tensions and sliding velocities for the primary pulley contact region are shown in Figures 6.2 and 6.3, respectively. For easier comparison, the tractive tension and sliding velocity are plotted against the non-dimensional wrap angle. Although the sliding velocity is not a solution variable, it may be calculated from the solution using Equation 3.12. In the primary pulley, the tractive tension remains constant for a large portion of the contact region. This indicates the presence of an adhesion region between wrap angles of 0.2 and 0.5, and the plot of sliding velocity confirms this, as the values are below the threshold velocity for adhesion in the friction model. Decreasing the value of the friction coefficient in this region prevents the magnitude of the sliding velocity from decreasing further and allows the solver to converge without modifying the equations. The largest magnitudes of velocity are seen at the entrance and exit regions, as the belt seats and unseats in the pulley groove.

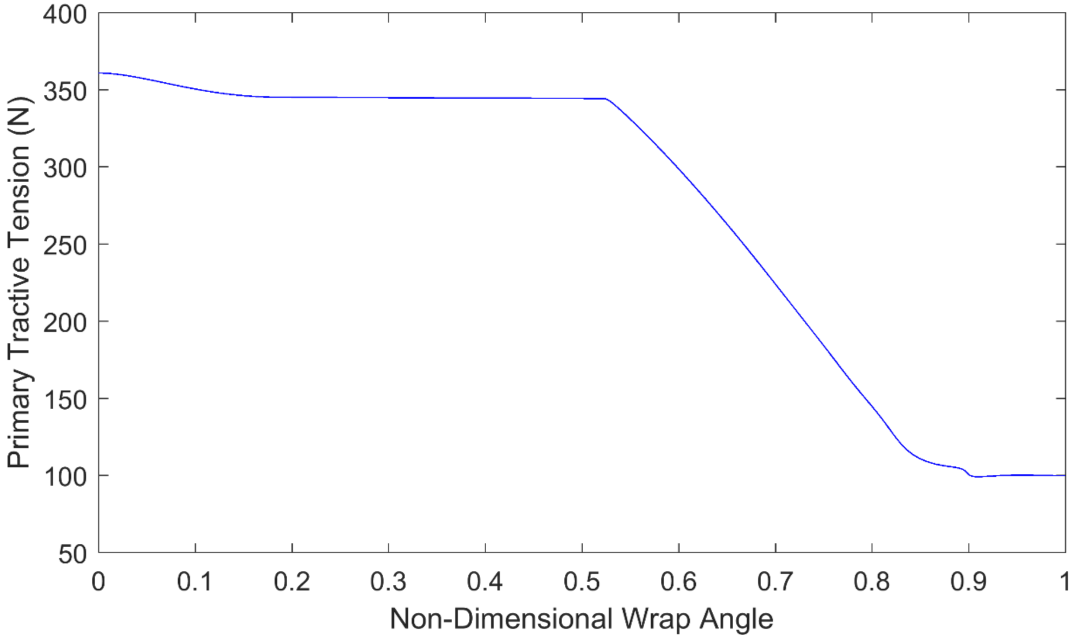


Figure 6.2. Plot of the tractive tension in the primary pulley contact region. The total wrap angle is 183.4°.

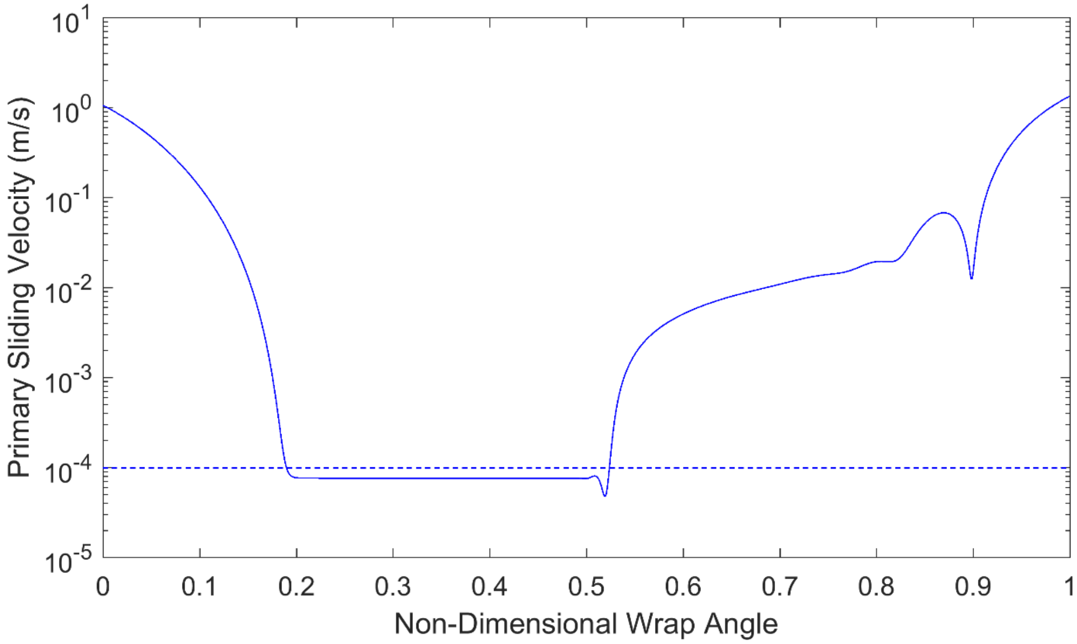


Figure 6.3. Plot of sliding velocity magnitude vs wrap angle in the primary pulley contact region. The dashed line indicates the threshold velocity for an adhesion region as used in the friction model. The friction model prevents the magnitude of the velocity from decreasing to zero in the adhesion region. Inflection points in the belt path and numerical effects from the solver cause rapid fluctuation of the velocity in the exit region.

Plots of the tractive tension and sliding velocity for the secondary pulley contact region are seen in Figures 6.4 and 6.5, respectively. The increase in tension for the secondary pulley is more gradual than that of the primary pulley, and there are no regions in the plot where the tension remains constant. As seen in the graph of sliding velocity, the magnitude never decreases below the threshold of 10^4m/s, defined in the friction model. This indicates that there is not an adhesion region present in this pulley.

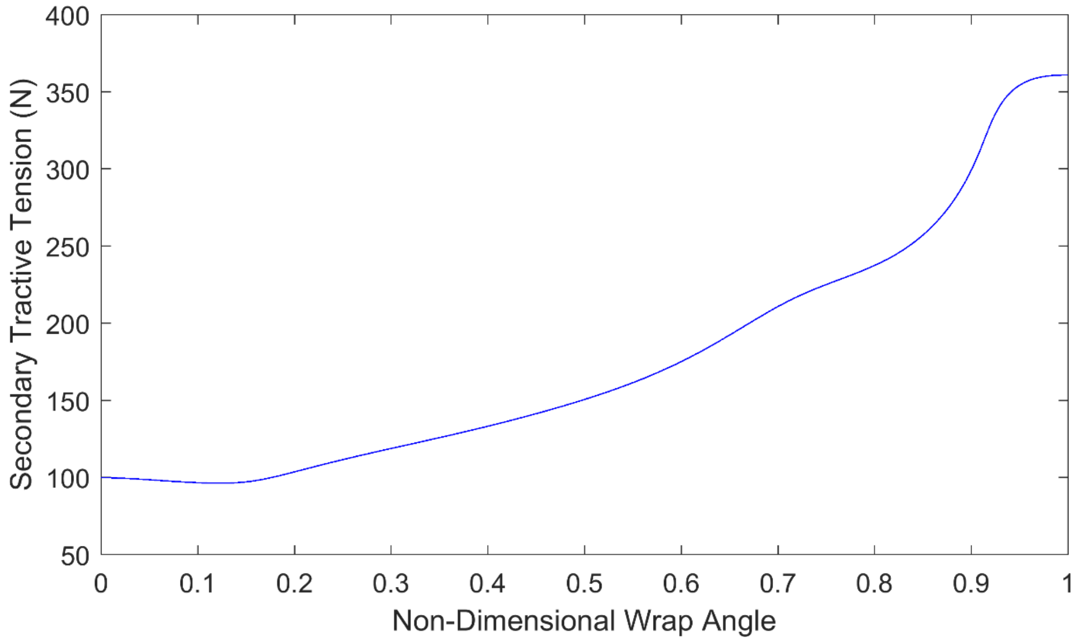


Figure 6.4. Plot of the tractive tension in the secondary pulley contact region. The total wrap angle is 183.9°.

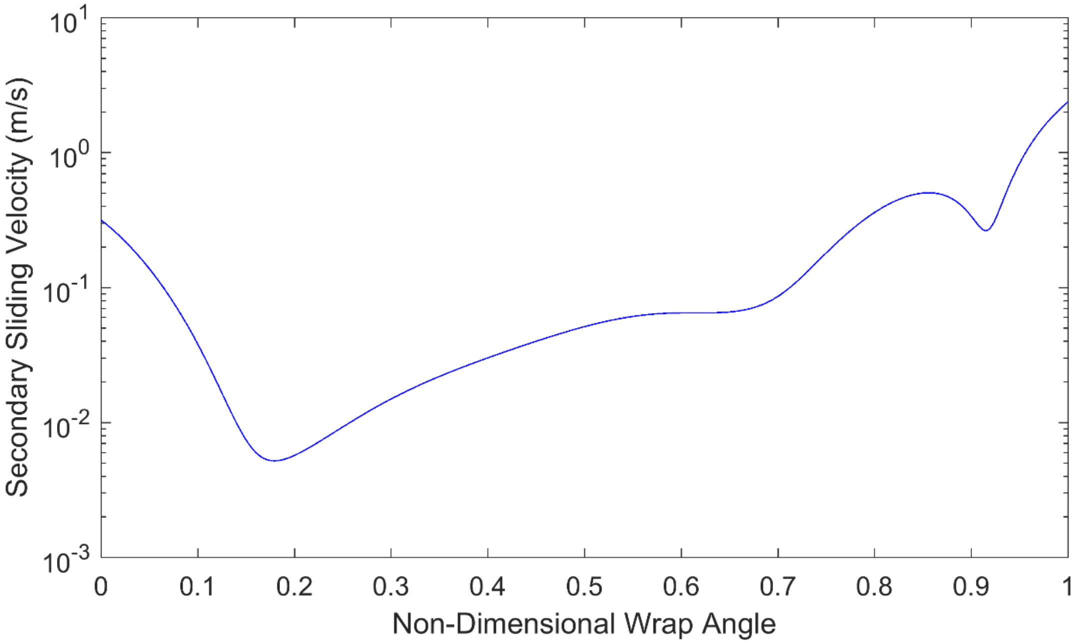


Figure 6.5. Plot of sliding velocity vs wrap angle in the secondary pulley contact region. The magnitude of the velocity does not decrease below the threshold of 10^{-4}m/s, indicating the non-existence of an adhesion region in this pulley.

Plots of the radial penetration for the primary and secondary pulleys are shown in Figures 6.6 and 6.7. For the primary pulley, the belt enters the groove at the tight span tension, leading to higher seating initially than that of the secondary pulley, where the belt enters at the slack span tension. More insight into the direction of the friction forces acting on the belt is given by calculating the sliding angle, γ. This quantity is also not a solution variable, but may be determined from the solution variables using Equation 3.13. From the definition of the sliding angle, a value of 0° corresponds to a sliding velocity pointing radially outward, and a value of 180° corresponds to a velocity pointing radially inward. Equation 3.13 calculates the angle from the tangent of the radial and circumferential components of the sliding velocity. The directions of these values are needed to confirm that the correct root of the inverse tangent is chosen. A positive angle of inclination indicates a sliding angle between 0° and 90° or 270° and 360°, where a negative angle of inclination indicates a sliding angle between 90° and 270°. Plots of the sliding angle for the primary and secondary pulleys are shown in Figures 6.8 and 6.9.

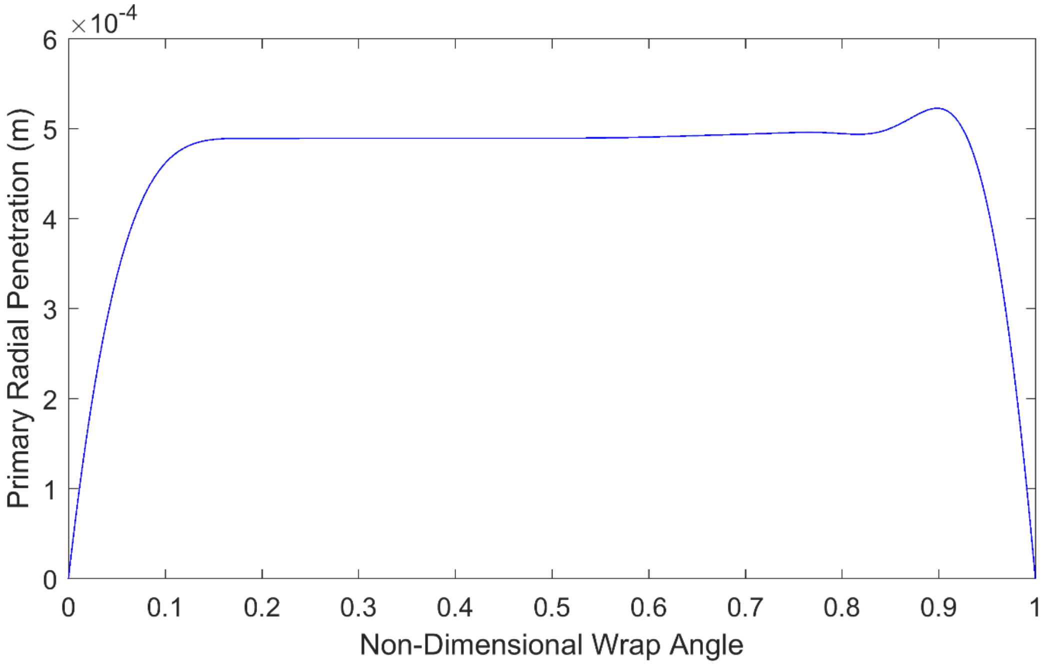


Figure 6.6. Plot of radial penetration vs wrap angle for the primary pulley contact region. The belt enters the pulley at the tight span tension, causing large values initially, and friction keeps the belt seated at this value until it exits the pulley.

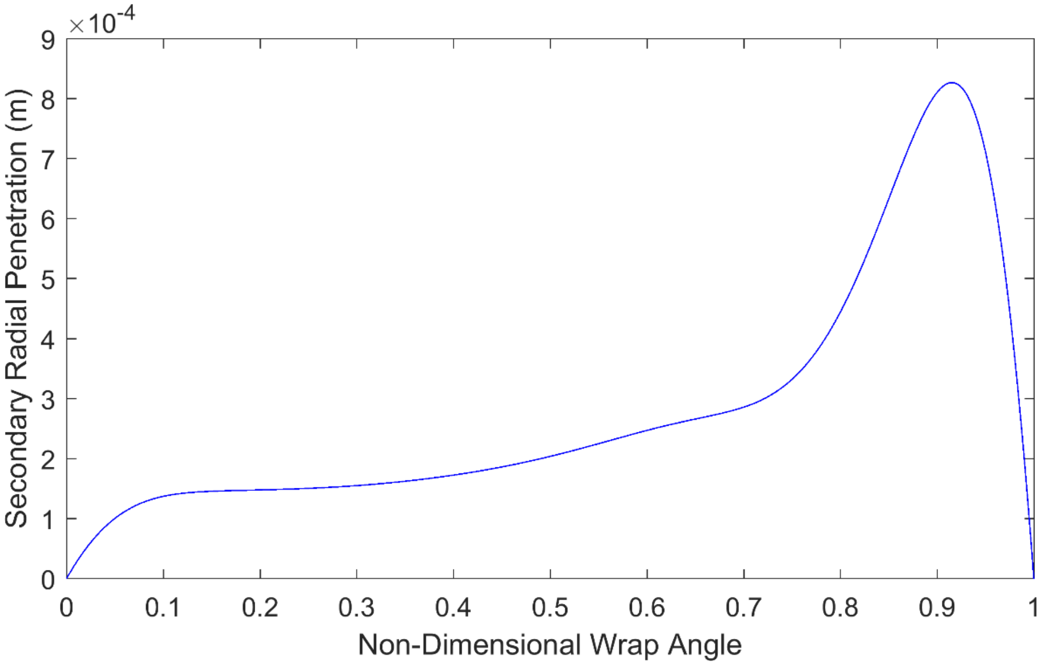


Figure 6.7. Plot of radial penetration vs wrap angle for the secondary pulley contact region. The belt enters the pulley at the slack span tension, causing low values initially that increase as tension increases, until the belt exits the pulley.

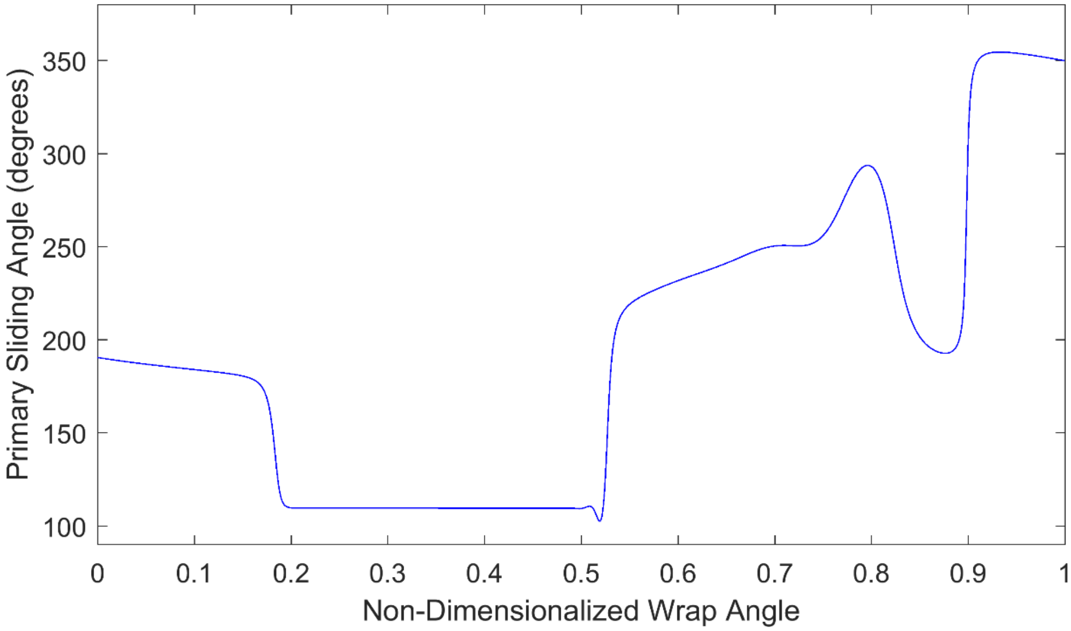


Figure 6.8. Plot of the sliding angle vs wrap angle for the primary pulley. Angles of 0°/360° indicate a sliding velocity pointing radially outward, and an angle of 180° indicates a sliding velocity radially inward.

Sliding angles between 90° and 270° indicate seating, and the plot for the primary pulley indicates that belt seating continues for approximately the first 75% of the contact region, while most of the unseating does not occur until the last 10% of the contact region. A similar trend is seen in the plot of the secondary pulley sliding angle, shown in Figure 6.9. No unseating occurs in the secondary pulley until after the first 90% of the contact region. Plots of the belt angle of inclination for the primary and secondary pulleys are shown in Figures 6.10 and 6.11, respectively. These plots show similar trends in seating and unseating with negative angles indicating seating, and positive angles indicating unseating. For both pulleys, the largest angles are seen in the entry and exit regions. The entry and exit angles for the primary are approximately -2.48° and 3.15°, while the corresponding angles for the secondary pulley are -0.75° and 5.65°.

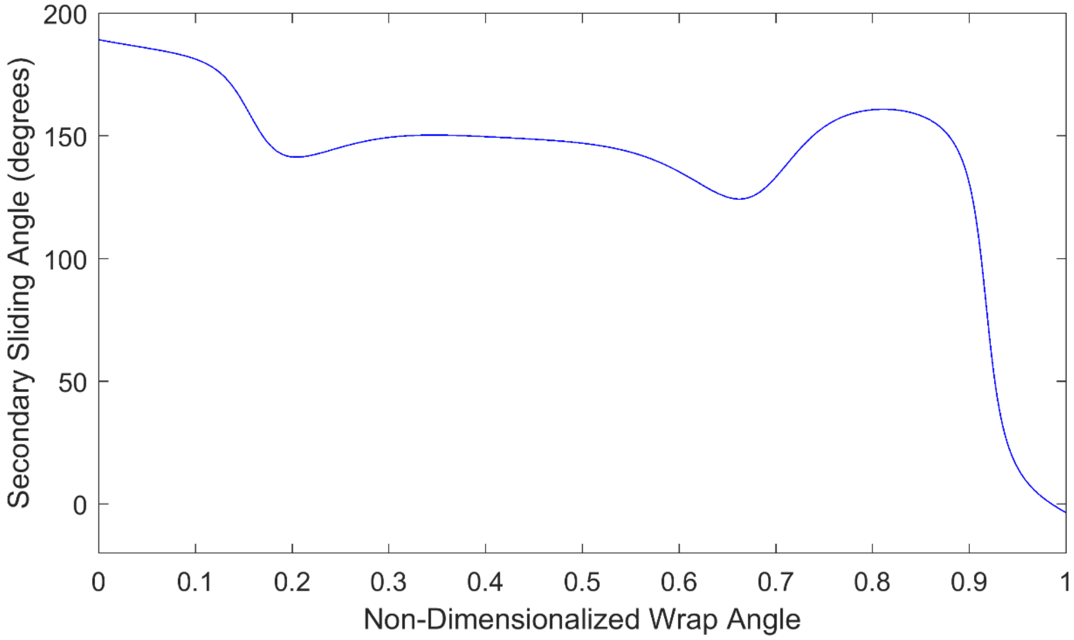


Figure 6.9. Plot of the sliding angle vs wrap angle for the secondary pulley. An angle of 0° indicates a sliding velocity pointing radially outward, and an angle of 180° indicates a sliding velocity radially inward.

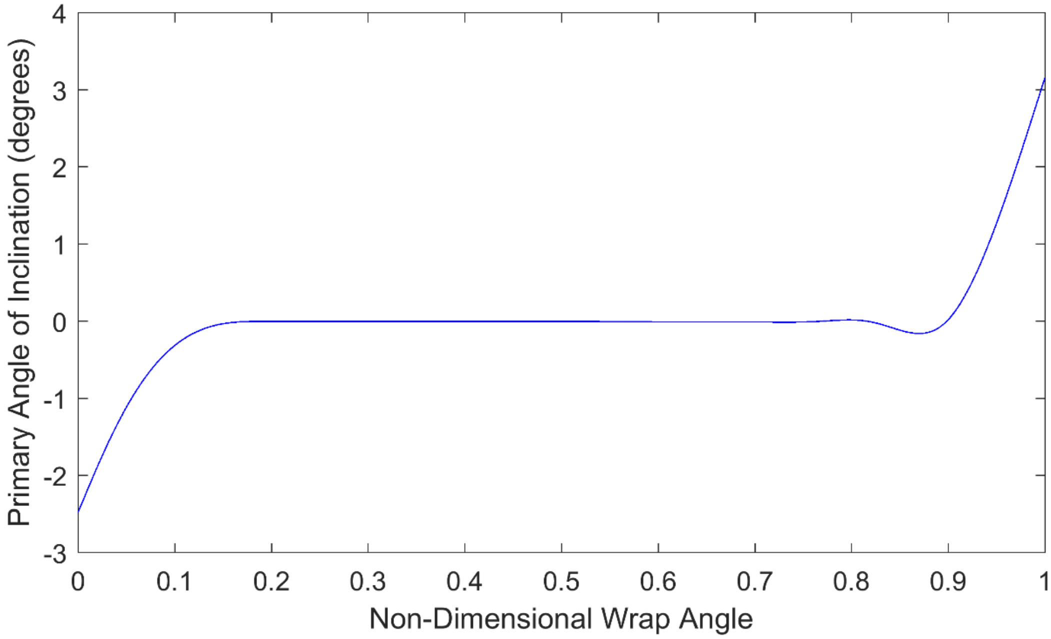


Figure 6.10. Plot of the angle of inclination vs wrap angle for the primary pulley. The largest magnitude angles are seen in the entry and exit regions, with negative angles indicating seating, and positive angles indicating unseating.

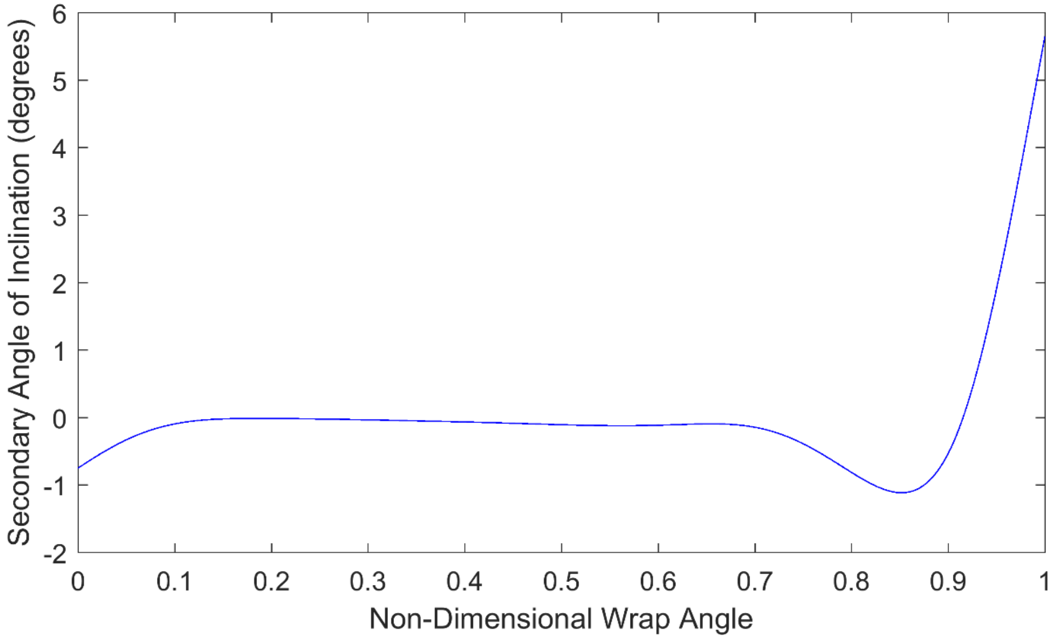


Figure 6.11. Plot of the angle of inclination vs wrap angle for the secondary pulley. Negative angles indicate rapid belt seating for non-dimensional warp angles between 0.7 and 0.9.

6.1.2 Axial Force Calculations

The axial force is integrated from the solution variables, and the value of this integral vs wrap angle is also determined in the solution using Equation 5.61. The axial force is the value of the integral at the end of the contact region. For the 1:1 case, these values are 934.3N for the primary pulley, and 573.2N for the secondary pulley. The derivative of the solution variable can be used to show the distribution of the axial pressure between the belt and sheave across the contact region. In addition to the values of the solution variables, the solution structure returned by bvp4c in MATLAB also includes the values of the derivatives of the solution variables at the mesh points. Equations 3.4 and 5.2 are then used to convert the derivative with respect to non-dimensional wrap angle that is returned in the solution to a derivative with respect to arc length as follows:

$$\frac{dA}{ds} = \frac{\cos(\phi)}{R - x}\,\frac{1}{\Phi}\frac{dA}{d\hat{\phi}} \tag{6.2}$$

This derivative with respect to arc length has units of N/m, and is the force per length exerted on the sheave by the belt. Plots of these derivatives for the primary and secondary pulleys are shown in Figures 6.12 and 6.13, respectively.

The axial force per length may also be used to determine the location for the resultant force on the sheave from the belt. The position of the belt leads to a resultant force that is not coincident with the center of the pulley. This location is useful for the design of a CVT to determine the binding moment that will be present on the movable sheave for each pulley. The magnitude of the resultant force is equal to the magnitude of the axial force, and the x and y coordinates of the resultant are determined relative to the pulley centers through the following equations:

$$\bar{x}_A = \frac{\int_0^1 \hat{A}' x_A \, d\hat{\phi}}{\int_0^1 \hat{A}' \, d\hat{\phi}} = \frac{\int_0^1 \hat{A}' x_A \, d\hat{\phi}}{A} \tag{6.3}$$

$$\bar{y}_A = \frac{\int_0^1 \hat{A}' y_A \, d\hat{\phi}}{\int_0^1 \hat{A}' \, d\hat{\phi}} = \frac{\int_0^1 \hat{A}' y_A \, d\hat{\phi}}{A} \tag{6.4}$$

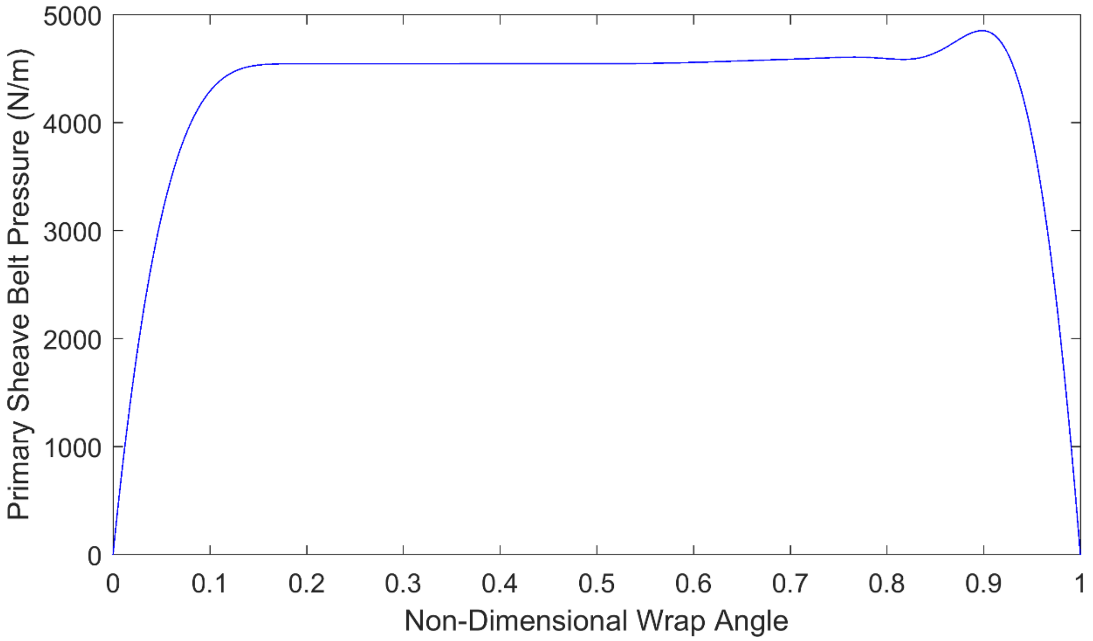


Figure 6.12. Belt axial force per arc length for the primary pulley contact region.

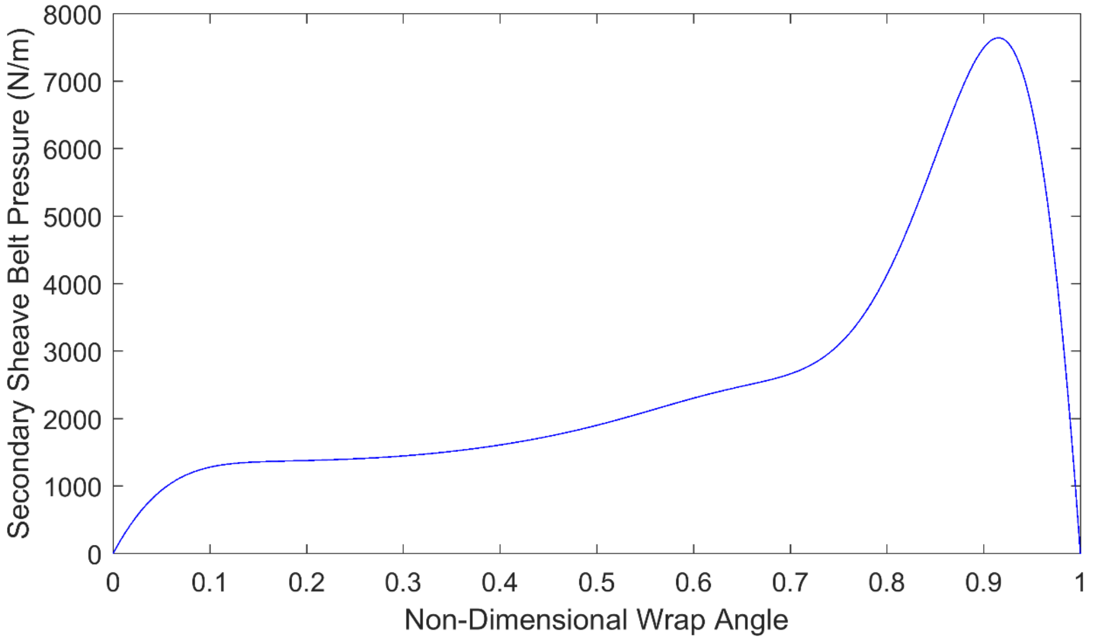


Figure 6.13. Belt axial force per arc length for the secondary pulley contact region.

$$x_{AP} = (R_P - x_P)\cos(C_P + \Phi_P\hat{\phi}_P)$$
$$y_{AP} = (R_P - x_P)\sin(C_P + \Phi_P\hat{\phi}_P) \tag{6.5}$$

$$x_{AS} = (R_S - x_S)\cos(\pi - C_S - \Phi_S + \Phi_S\hat{\phi}_S)$$
$$y_{AS} = (R_S - x_S)\sin(\pi - C_S - \Phi_S + \Phi_S\hat{\phi}_S) \tag{6.6}$$

Right hand sums are used to evaluate the integrals from the solution vector. The location of the resultant force on the primary pulley is 44.9mm to the left and 1.4mm below the pulley center. This is illustrated in Figure 6.14. The resultant force on the secondary pulley is located 36.2mm to the right and 29.5mm above the pulley center. These resultants create binding moments of 41.9N*m for the primary pulley and 26.8N*m for the secondary pulley.

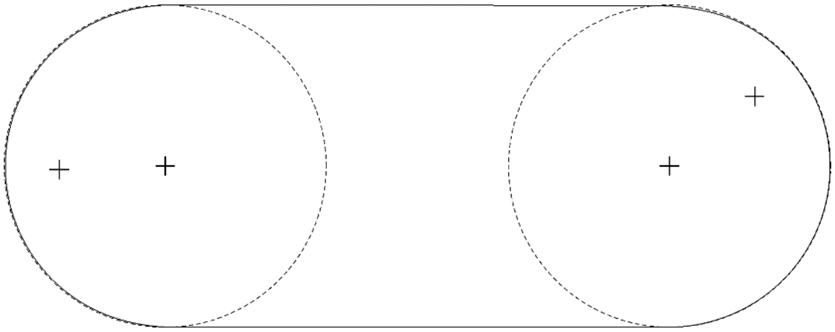


Figure 6.14. Location of the resultant forces from the belt distributed axial loads on the sheaves. The primary is shown on the left, and the secondary on the right, with both rotating counter-clockwise. The resultant for the primary pulley is located 44.9mm from the pulley center, and the resultant on the secondary pulley is located 46.7mm from the pulley center.

6.1.3 Span Results

Figures 6.15 and 6.16 show the belt paths in the tight and slack spans, respectively. Some curvature is present at the edges of the spans as the belt enters and exits the pulley. The belt tractive tensions in the spans are plotted in Figures 6.17 and 6.18.

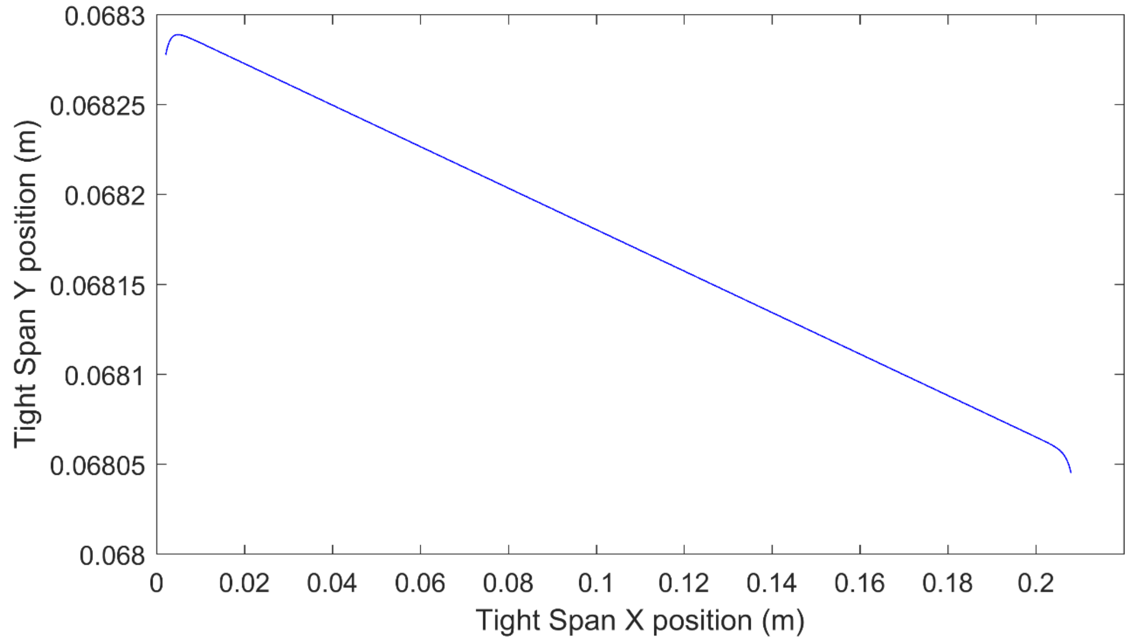


Figure 6.15. Belt trajectory in the tight span.

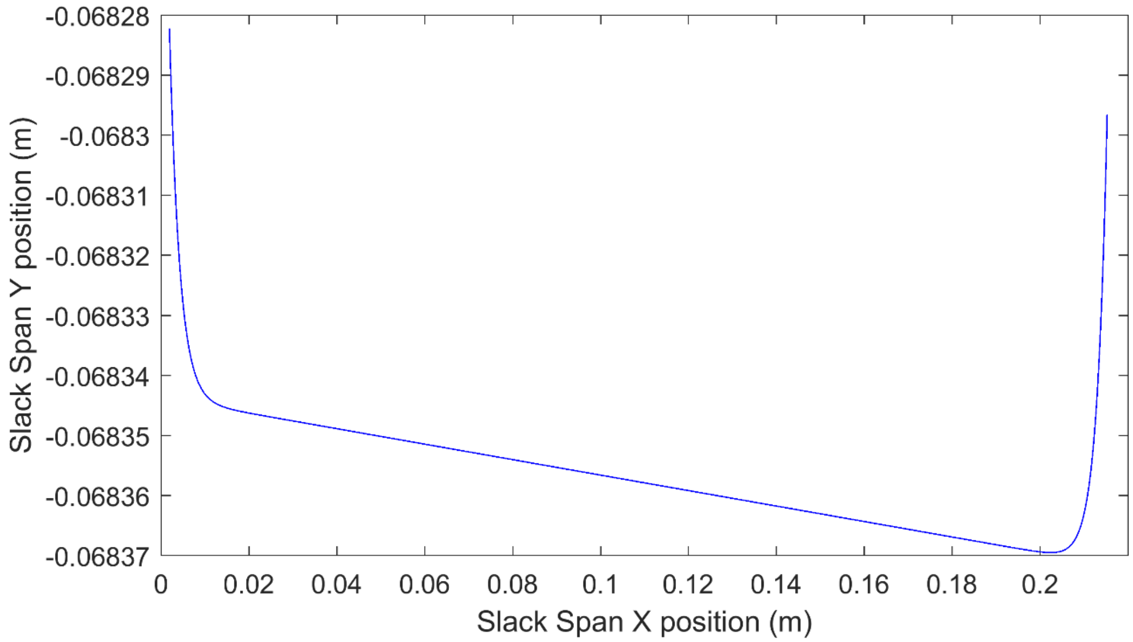


Figure 6.16. Belt trajectory in the slack span.

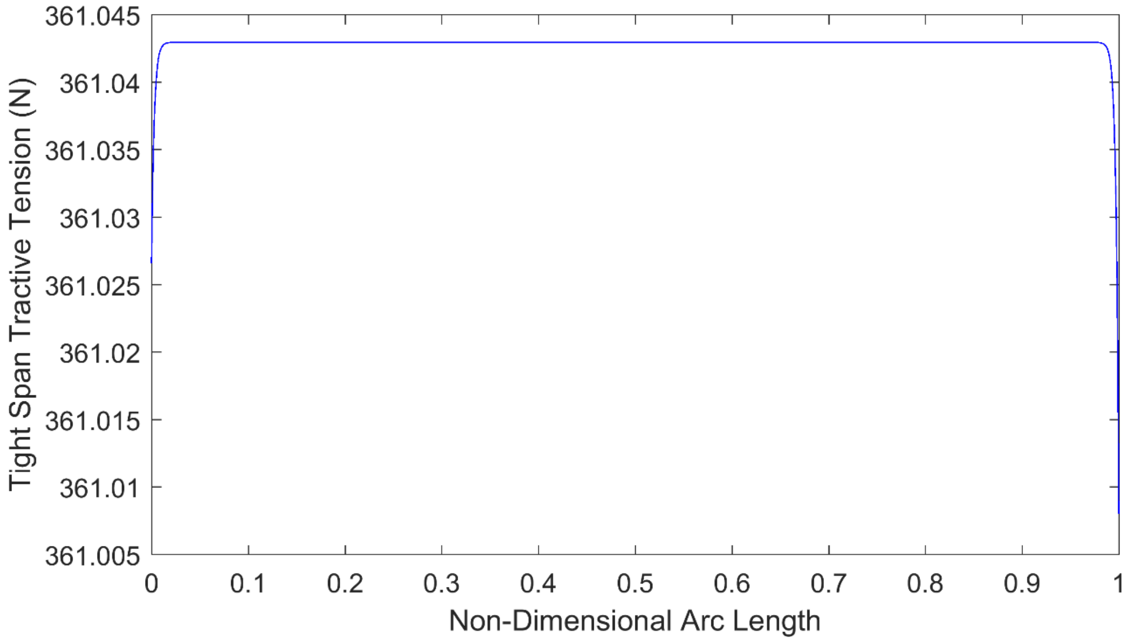


Figure 6.17. Tight span tractive tension. The value is calculated from the primary pulley moment boundary condition. This value changes at the edges of the span as the belt curvature changes.

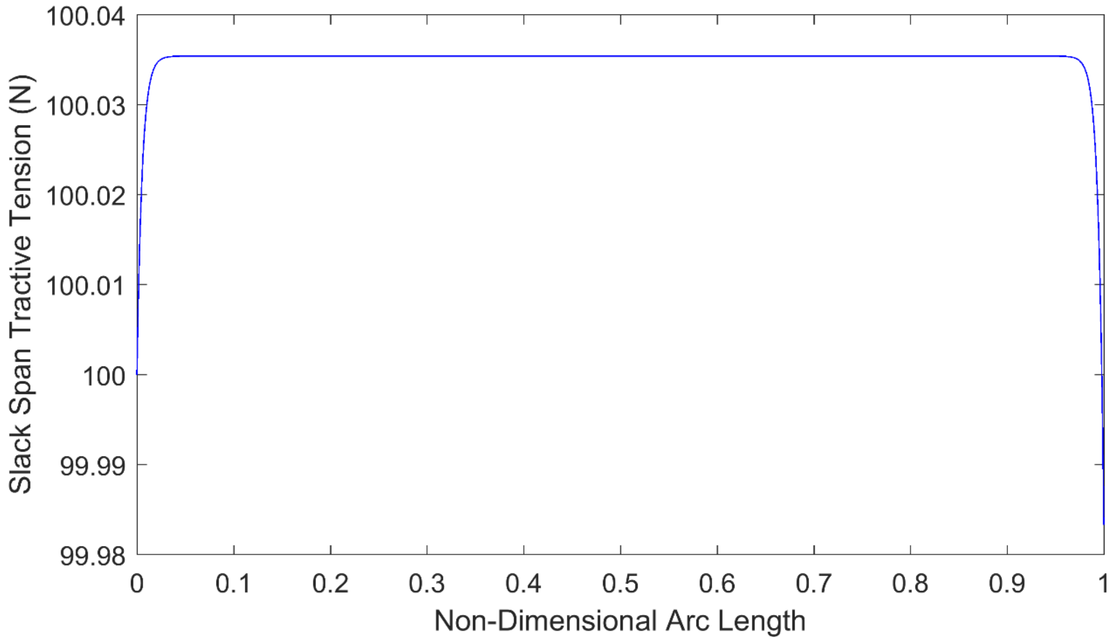


Figure 6.18. Slack span tractive tension. A boundary condition is applied at the exit of the primary pulley, shown on the left, to enforce this value at 100N.

6.1.4 Efficiency Calculations

The three efficiency components for a V-belt drive are speed, torque, and overall. Speed and torque efficiency are determined through the pulley geometry. Speed efficiency is defined using the following equation:

$$\eta_{speed} = \frac{\omega_S}{\frac{R_P}{R_S}\omega_P} \tag{6.7}$$

The quantity $\frac{R_P}{R_S}$ is the geometric gear ratio and would be ideally equal to the ratio of angular velocities, $\frac{\omega_S}{\omega_P}$. Belt slipping and elasticity effects cause the secondary angular velocity to be less than the ideal value, calculated from the given radii and primary pulley angular speed. The torque efficiency is calculated in a similar manner as follows:

$$\eta_{torque} = \frac{\left(\frac{\tau_S}{\tau_P}\right)}{\left(\frac{R_S}{R_P}\right)} \tag{6.8}$$

Effects of belt seating and unseating cause the secondary pulley torque to be less than the ideal value, calculated from the ratio of the pulley radii and the primary pulley torque boundary condition. The overall efficiency is the product of the speed and torque efficiencies, and is equal to the ratio of power out to power in for the system:

$$\eta_{overall} = \frac{\tau_S\ \omega_S}{\tau_P\ \omega_P} = \eta_{speed}\ \eta_{torque} \tag{6.9}$$

For the 1:1 ratio case, the speed, torque, and overall efficiencies are 99.39%, 90.67%, and 90.11%, respectively. The lack of an adhesion region in the secondary pulley does not seem to cause excessive speed loss in the system. The minimum sliding velocity in the secondary pulley is 0.0052m/s. Using the local belt radius and sheave angular velocity, the absolute velocity of the sheave in the circumferential direction is 24m/s at that location. This indicates that the dominant effect when calculating speed loss is belt elongation, rather than belt slipping.

6.2 Results for the Operating Range Radius Sweep

The 1:1 ratio case may be used as a starting point to extend the solution to other points in the CVT's operating range. Geometry definitions are established to approximate the relationship of the primary and secondary pulley nominal radii. Continuation methods are used to modify the radii of the pulleys from the 1:1 ratio case. Results of interest over the operating range are the efficiency and axial forces. Other results, such as the sliding velocity and tension over the contact region, provide more insight into the belt mechanics over the operating range.

6.2.1 Geometry Definitions

To find the relationship between the pulley radii, the same assumptions are made as the 1:1 ratio case. It is theoretically possible to treat one of the pulley radii as another unknown constant, and use the total belt unstretched length as an additional boundary condition to solve for it. However, the equations are especially unstable with respect to a change in radius, making this impossible in practice. Iteration for the pulley radius in the solution to match the belt unstretched length is another option, but this would result in a substantial increase in computation time due to the small increments required for continuation and the resulting number of steps needed. For simplicity, to relate the pulley radii during a sweep of the operating range, belt radial penetrations and belt extension in the longitudinal direction are assumed to be small, resulting in an approximate belt path that follows a constant radius in the contact regions, and follows a line tangent to both pulleys in the spans. This is illustrated in Figure 6.19.

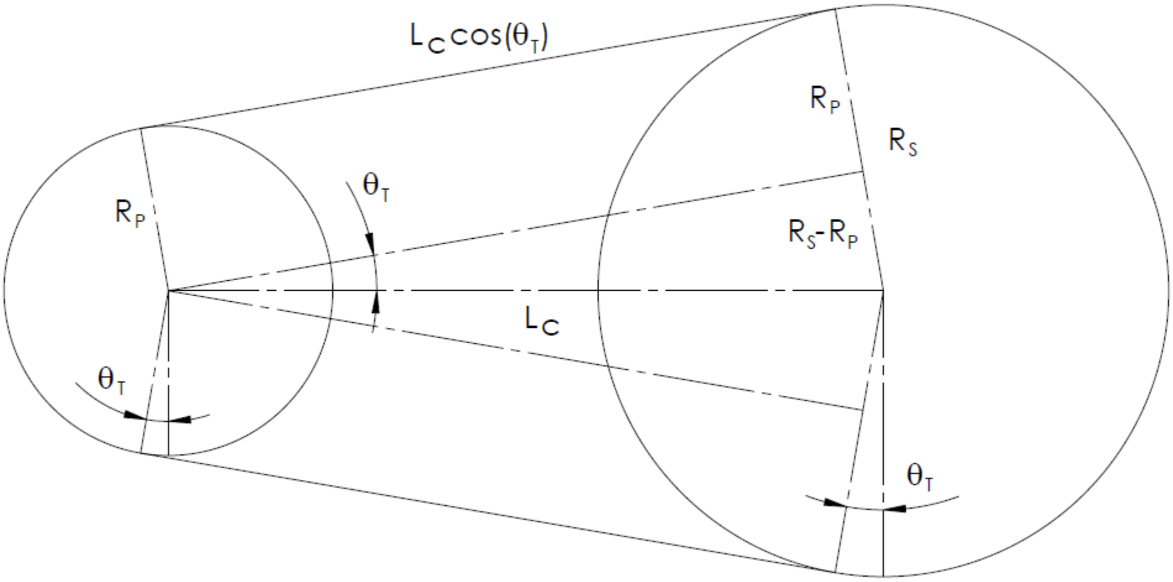


Figure 6.19. Simplified belt path used to relate the primary and secondary pulley radii. It is assumed that the contact regions form arcs at the nominal radii and the spans follow lines tangent to both pulley nominal radii. An angle, θ_T, is defined between the entrance/exit radii and the vertical.

The angle, θ_T, is determined using the following equation:

$$\sin^{-1}(\theta_T) = \frac{R_S - R_P}{L_C} \tag{6.10}$$

This angle approximates the pulley wrap angles as follows:

$$\begin{aligned} \Phi_\mathrm{P} &\approx \pi - 2\,\theta_T \\ \Phi_S &\approx \pi + 2\,\theta_T \end{aligned} \tag{6.11}$$

These quantities are used to find an expression relating the belt pitch length and the nominal radii:

$$\mathrm{L_P} - R_P\,\Phi_P - R_S\,\Phi_S - 2\,L_C\cos(\theta_T) = 0 \tag{6.12}$$

It is not possible to rearrange this equation to obtain an explicit expression for one radius in terms of the other. The equation must be solved numerically, and the MATLAB function used for this is given in the Appendix.

6.2.2 Solution Approach

Through trial and error, it is determined that some values are more easily adjusted through continuation than others. The system exhibits poor stability with respect to a change in radius, while other variables, such as bending stiffness, are more easily adjusted through continuation. Increasing the belt bending stiffness to an artificially high value results in a smother solution, with a significantly reduced number of mesh points. This increases the stability of the system of equations, decreases computation time, and allows larger step sizes to used for the radius through continuation, making it possible to sweep the pulley radii through the desired operating range in a reasonable amount of time. The solution structure is saved at regular intervals to result in an array of solutions at a range of operating radii. These solutions are then passed individually through a continuation loop to return the belt bending stiffness to the measured value. If further changes to the solution parameters are desired, such as changes in slack span tensions, these are easily obtained by individually adjusting the solutions in the array. The range of radii obtained through this method is illustrated in Figure 6.20, with the minimum and maximum reductions at 1.74:1 and 0.72:1, respectively.

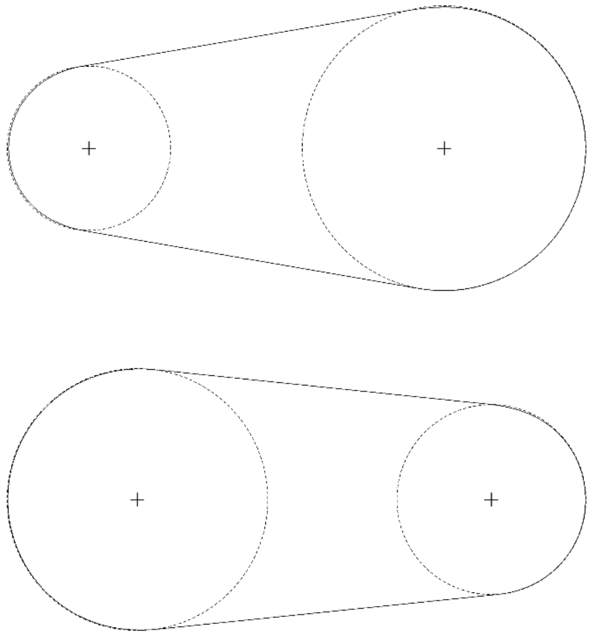


Figure 6.20. Range of pulley radii obtained in the solution. The primary pulley is located on the left. The lowest reduction ratio obtained is 1.74:1 and the highest ratio is 0.72:1 calculated from the pulley nominal radii.

6.2.3 Axial Force Calculations

A plot of the axial forces for both pulleys through the operating range is shown in Figure 6.21. Similar to the 1:1 ratio case, the axial force in the primary is always larger than the secondary. Both forces decrease as the system approaches high range, with the axial force in the secondary seeing a larger decrease in magnitude. Distributions of axial belt pressure are shown in Figures 6.22 and 6.23 for the primary and secondary, respectively. Figure 6.22 indicates that as the system approaches high range, the axial belt pressure decreases in the primary. The primary pulley moment boundary condition is a constant, forcing the magnitude of the tight span tension to decrease as the radius increases. This effect combines with the associated increase in wrap angle to decrease the pressure as the system approaches high range. Figure 6.23 indicates that the axial belt pressure does not exhibit a strong relationship with operating point for the secondary pulley. As the system approaches high range, the decrease in tight span tension couples with the decrease in wrap angle to hold the belt pressure approximately constant.

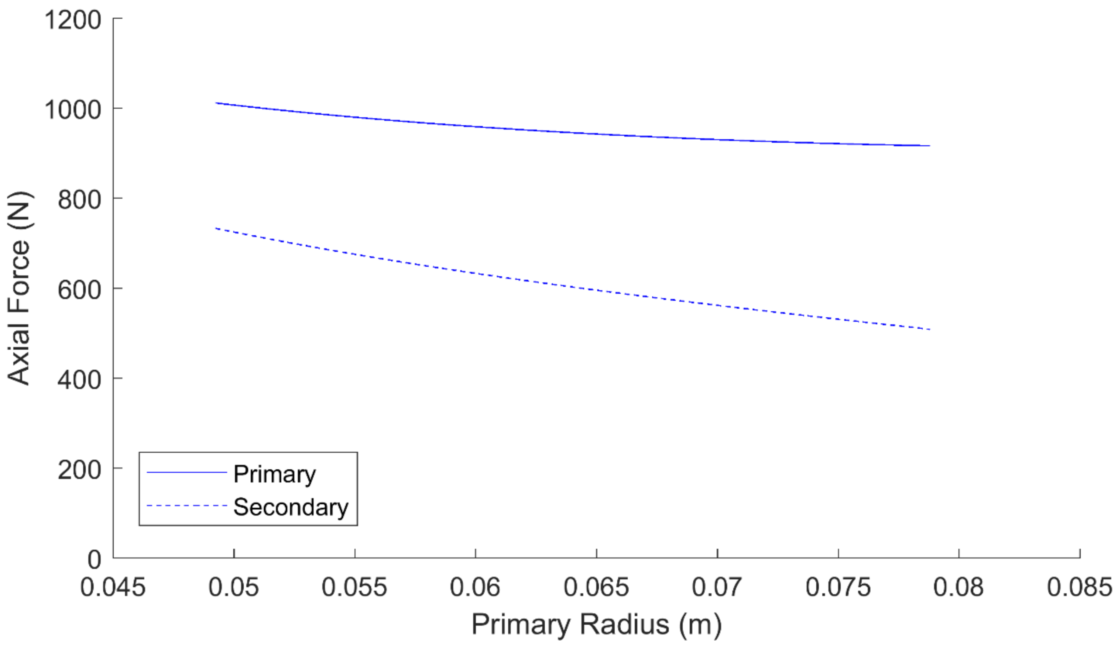


Figure 6.21. Plot of axial forces in both pulleys over the operating range. This corresponds to a slack span tractive tension of 100N, with the lowest reduction on the left. Both axial forces decrease as the CVT moves toward high range, and the difference between the forces in the two pulleys also increases.

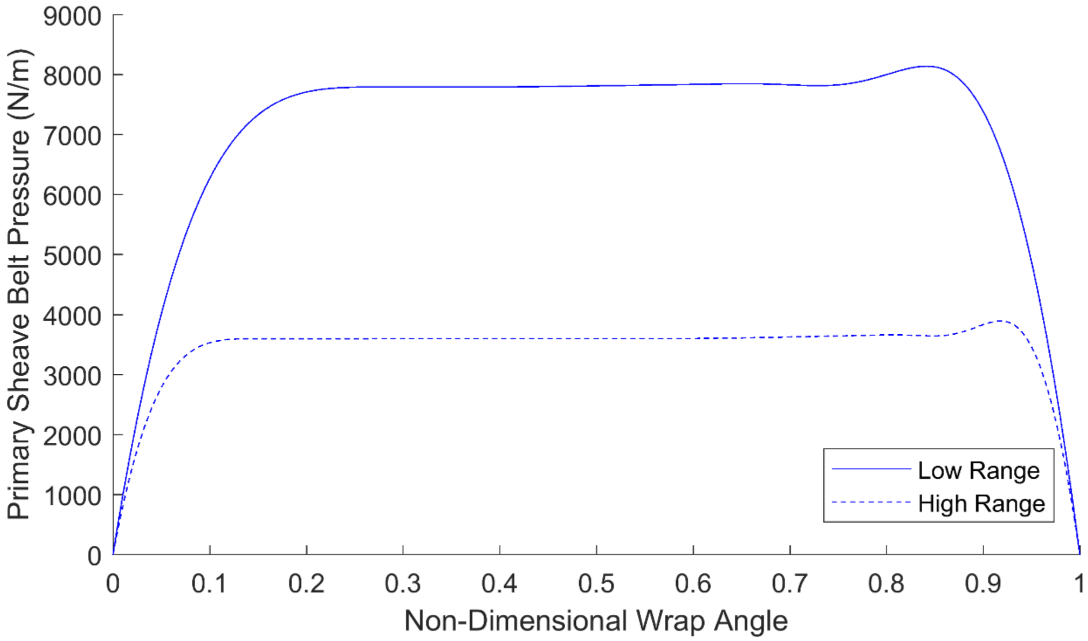


Figure 6.22. Axial belt pressure in the primary pulley for the edges of the operating range. The belt pressure decreases as the system approaches high range.

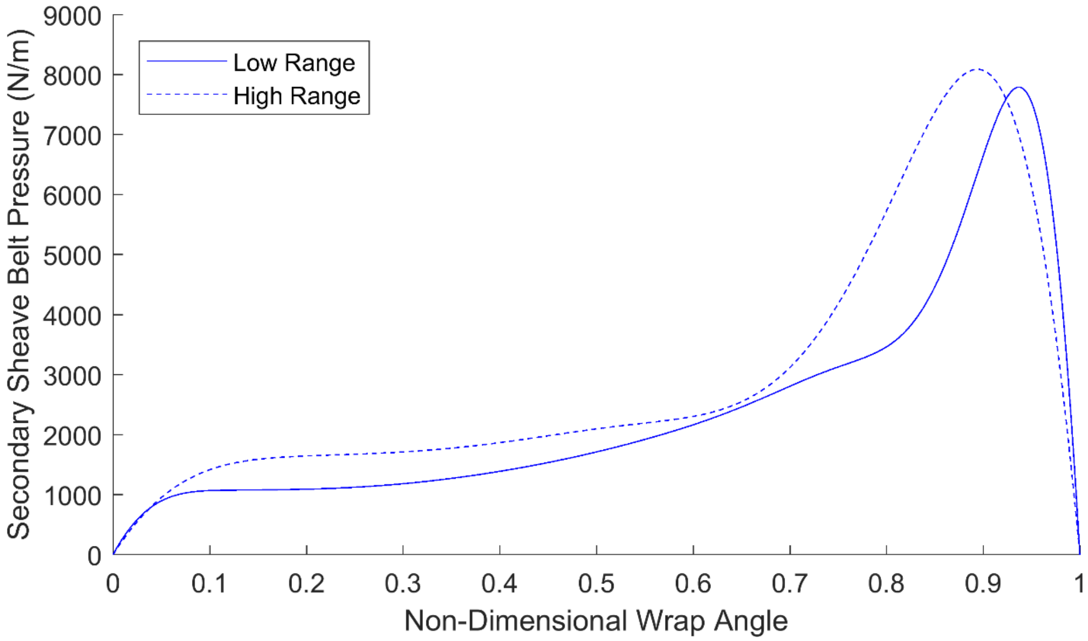


Figure 6.23. Axial belt pressure in the secondary pulley for different points in the operating range. The belt pressure does not show a strong relationship between operating point, exhibiting similar magnitudes across the operating range.

6.2.4 Efficiency Calculations

The efficiencies of the system, including speed, torque, and overall efficiency over the operating range, are plotted in Figure 6.24. For all points in the operating range, the power losses due to speed are small compared to those due to torque. The magnitude of the torque losses increase steadily as the system approaches high range, and the magnitude of the speed losses decrease steadily as the system approaches high range. Reynolds first discovered speed losses in flat belt drives due to belt extensibility in the contact region. As the belt changes length between the entrance and exit regions, a certain amount of belt slip in the circumferential direction is unavoidable, and results in speed loss [9]. In high range, the belt tractive tension in the tight span is at a minimum, causing these effects of belt extensibility to be minimized. Although the tension and belt radial penetration decrease as the system approaches high range, the torque efficiency continues to decrease. The decrease in tension may also cause the belt unseating to take place more slowly in the secondary, leading to larger wrap angles and a higher torque loss in the pulley.

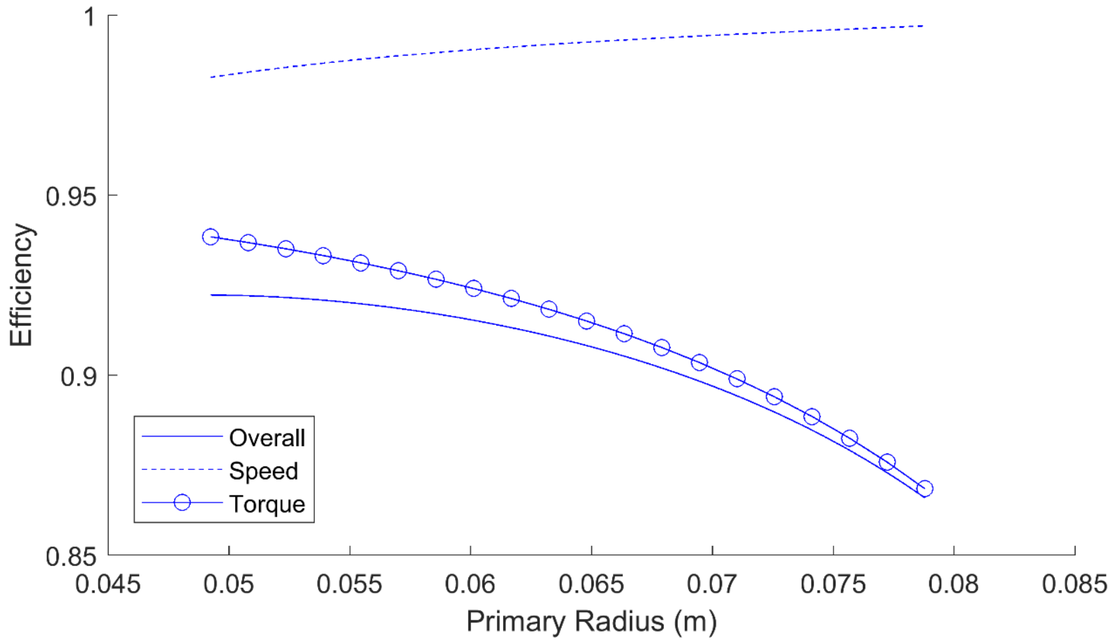


Figure 6.24. Plot of speed, torque, and overall efficiency over the operating range. Speed losses are at a maximum when the system is in low range, and torque losses are at a maximum when the system is in high range.

6.2.5 Additional Results

Additional results of interest over the operating range are the belt sliding velocities and tensions in the contact regions. Plots of the sliding velocities in high and low range for the primary and secondary pulleys, respectively, are shown in Figures 6.56 and 6.26. Increasing wrap angle and decreasing tight span tractive tension in the primary pulley leads to a more pronounced adhesion region in high range. The adhesion region comprises approximately 10% of the contact region in low range, compared to 40% in high range. In the secondary pulley, increasing wrap angle is also associated with an increase in tight span tractive tension, leading to the non-existence of an adhesion region in the pulley. The magnitude of the sliding velocities increase as the system approaches high range, with the minimum magnitudes for low and high range corresponding to approximately 10^{-3}m/s and 10^{-2}m/s, respectively.

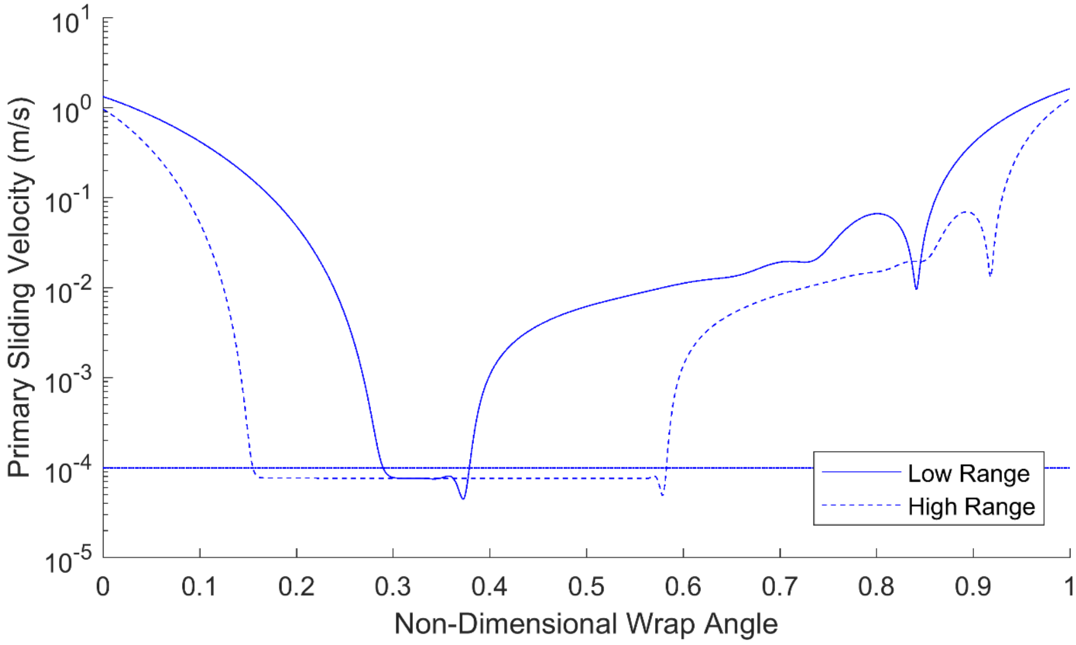


Figure 6.25. Plot of sliding velocity vs wrap angle over the operating range in the primary pulley. The threshold adhesion velocity of 10^{-4}m/s is also indicated. The adhesion region is more pronounced in the high range.

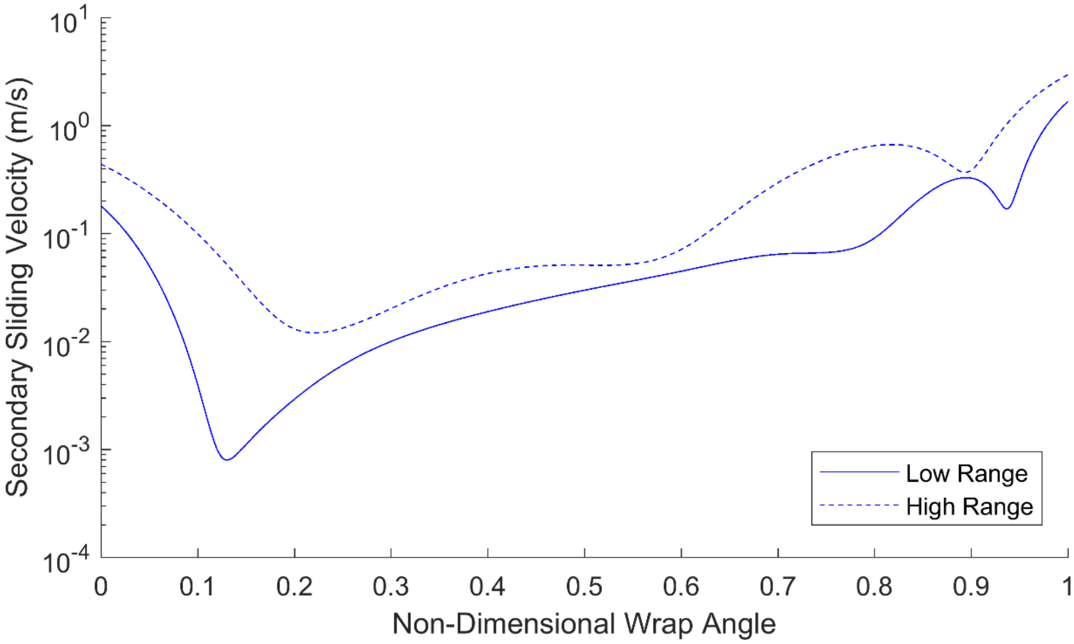


Figure 6.26. Plot of sliding velocity vs wrap angle over the operating range in the secondary pulley. Neither case resulted in sliding velocities below the threshold to form an adhesion region.

Plots of the tractive tension in the primary and secondary pulleys are shown in Figures 6.27 and 6.28, respectively. The tight span tension decreases from 460N in low range to 326N in high range. The tractive tension validates the appearance of adhesion regions in the primary, as seen in the sliding velocity plots. The effects of the exit region are more pronounced in low range, as the tension remains approximately constant for the last 20% of the contact region. Exit region effects are only seen in the last 10% of the contact region in high range. The tension profile in the secondary pulley is not as dependent on operating range. Both profiles show slight increases in tension over the first half of the contact region, and large increases in the last 10-20% of the region.

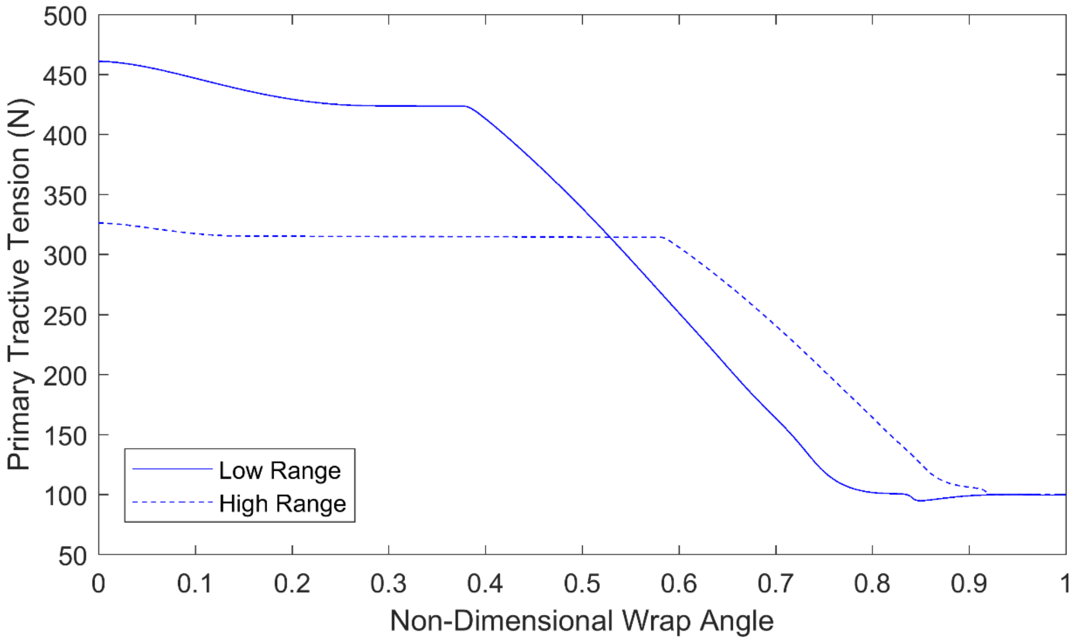


Figure 6.27. Tractive tension in the primary pulley contact region over the operating range..

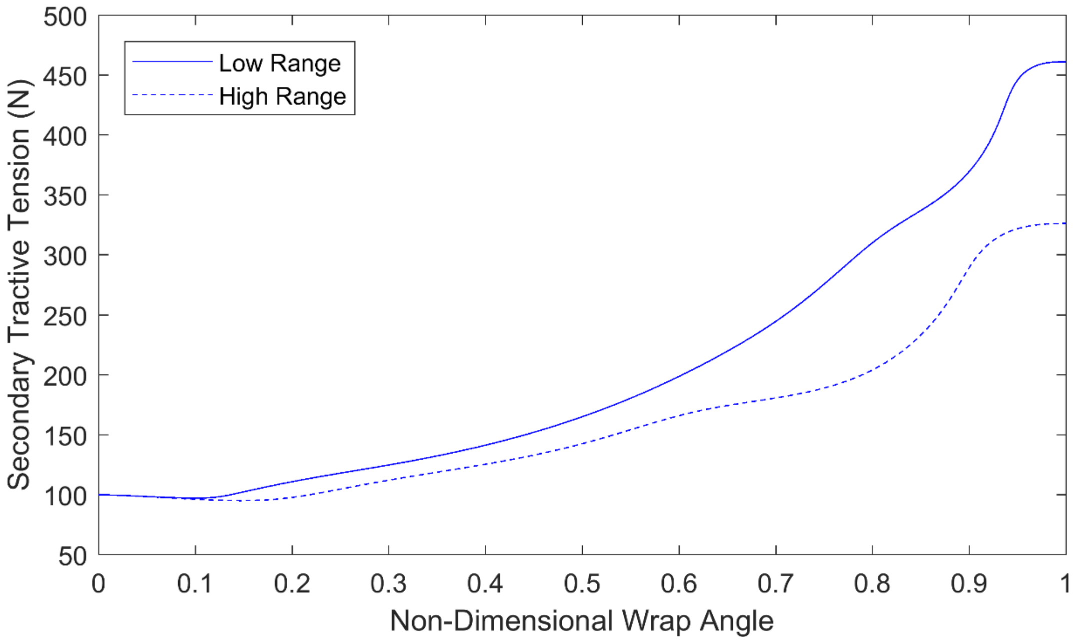


Figure 6.28. Tractive tension in the secondary pulley over the operating range.

7 Parameter Study on Efficiency

The choice of certain design parameters affect the efficiency of a CVT. Many qualitative assertions are made by Aaen regarding efficiency [4], but for application-specific design, more quantitative results are needed. Some parameters a designer has the ability to modify are sheave angle, belt stiffness, and slack span tractive tension through the adjustment of the axial force in the pulleys. Gerbert states that typical values for the kinetic friction coefficient range from 0.3 to 0.5 [7]. Since an experimentally determined value is not available, the effects of changing this parameter within this range are also considered.

7.1 Sheave Angle

Plots of the overall, speed, and torque efficiencies for varying sheave angles are shown in Figures 7.1, 7.2, and 7.3, respectively. Altering the sheave angle has several effects on the dynamics of the system. Since the belt lateral stiffness is determined in terms of radial spring stiffness, it is dependent on sheave angle. For the purposes of this analysis, it is assumed that the same belt is used, with the only change being the belt's wedge angle. This implies that the bending and axial stiffnesses of the belt remain constant. Equation 4.1 is used to find the belt radial spring stiffness from load cell data and is repeated here:

$$k = 4\frac{m}{L_S}\tan(\beta) \qquad (7.1)$$

It is assumed that the quantity, $\frac{m}{L_S}$, the slope of force vs displacement divided by the sample length, remains constant throughout the parameter sweep. The belt lateral stiffness may then be determined using the tangent of the sheave angle under consideration. The measured value for k at a 12° sheave angle is 18.71*10^{6}N/m^2. This value decreases to 12.37*10^{6}N/m^2 at an 8° sheave angle, and increases to 25.24*10^{6}N/m^2 at a 16° sheave angle. Increasing the sheave angle also increases the radial component of the normal pressure exerted by the sheave on the belt, and decreases the radial component of the friction between the belt and sheave by increasing the angle of the sliding plane.

These effects combine to reduce the belt seating effects and result in an increase in efficiency as the sheave angle increases. A table of the maximum belt radial penetration for both pulleys by sheave angle and operating point are shown in Table 7.1.

Table 7.1. Maximum belt radial penetration for various sheave angles and operating points.

Sheave Angle, β	Operating Point	Primary Pulley	Secondary Pulley
8°	Low Range	1.5mm	1.5mm
8°	High Range	0.74mm	1.5mm
12°	Low Range	0.89mm	0.84mm
12°	High Range	0.42mm	0.88mm
16°	Low Range	0.57mm	0.54mm
16°	High Range	0.27mm	0.56mm

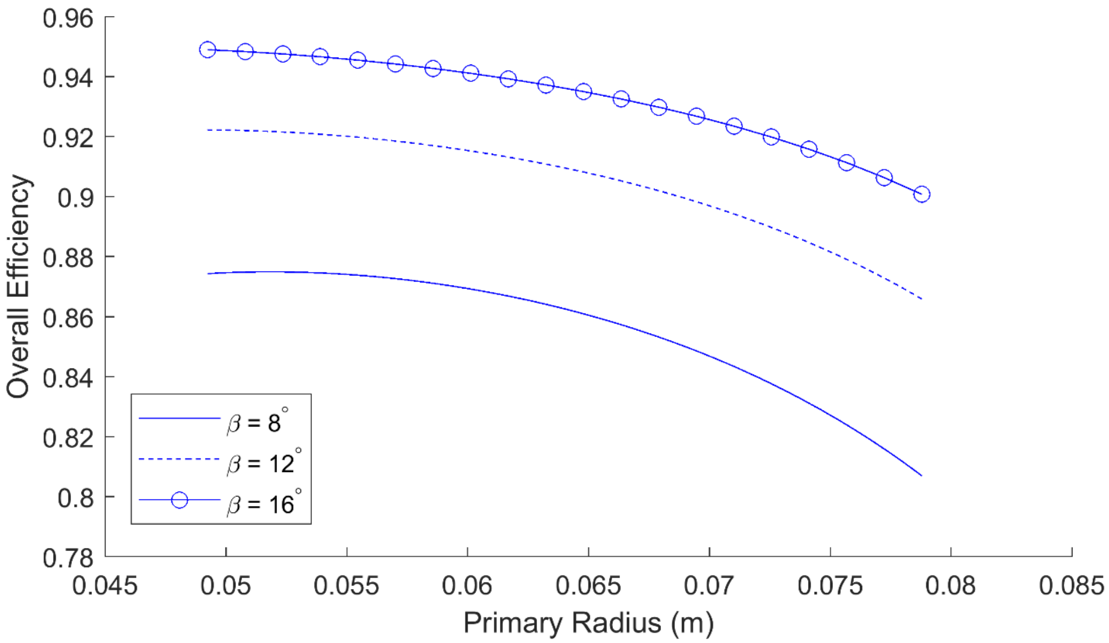


Figure 7.1. Plot of overall efficiency for varying sheave angles. Increasing the sheave angle increases the efficiency throughout the operating range.

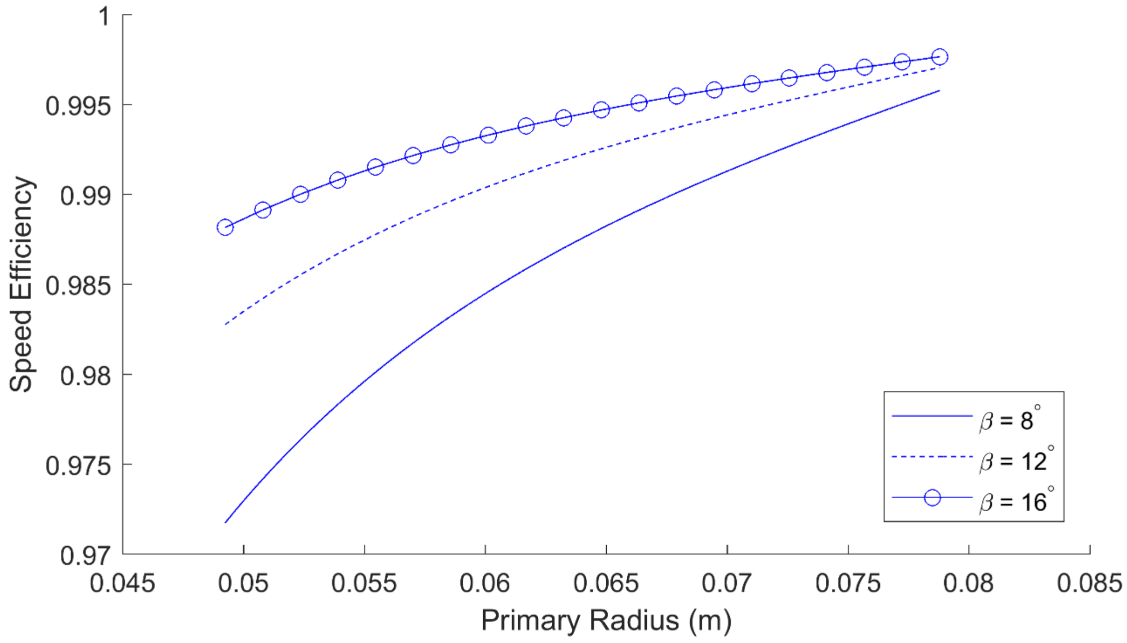


Figure 7.2. Plot of speed efficiency for varying sheave angles. Higher sheave angles show the largest improvement in efficiency in low range.

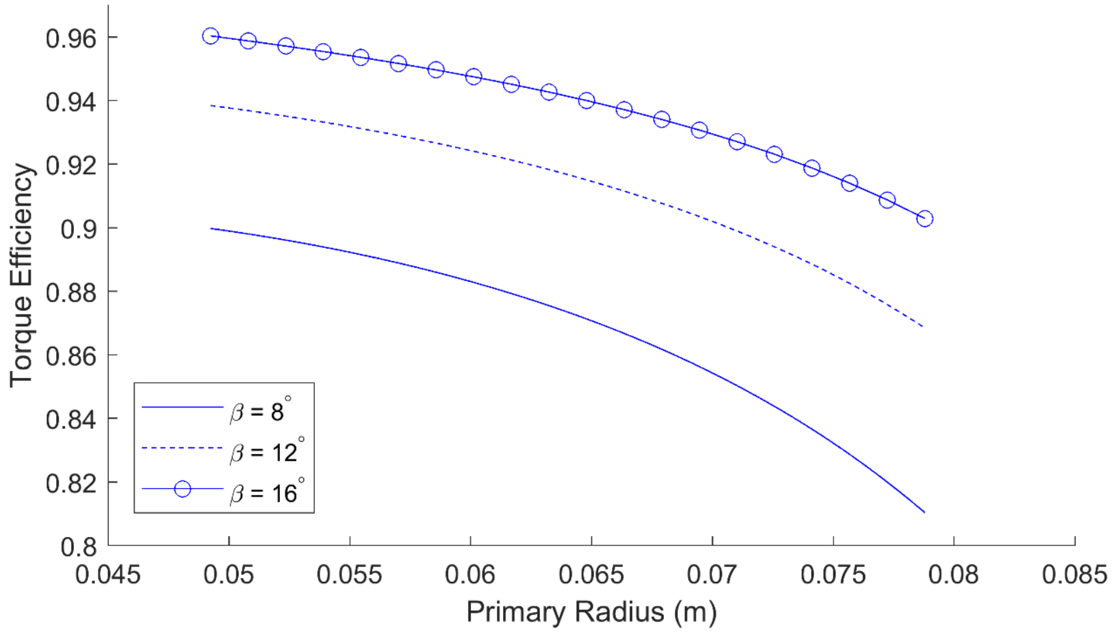


Figure 7.3. Plot of torque efficiency for varying sheave angles. Increasing the sheave angle shows an improvement in torque efficiency throughout the operating range.

7.2 Belt Bending Stiffness

Plots of the efficiencies vs belt bending stiffness are shown in Figures 7.4 to 7.6. Decreasing the belt bending stiffness shows substantial improvements in efficiency. Cases are plotted for stiffnesses of 0.025N*m^2, 0.2N*m^2, and 0.5N*m^2. In low range, decreasing the stiffness from 0.5N*m^2 to 0.025N*m^2 increases the efficiency by 7.4%, and in high range it improves the efficiency by 14.9%. Plots of the radial penetration for these cases in low and high range are shown in Figures 7.7 to 7.10. These plots show that increasing the belt bending stiffness causes smoothing effects on the belt path in the entrance and exit regions, causing higher belt radial penetrations with increased bending stiffness. The largest effects are seen in the secondary pulley exit regions.

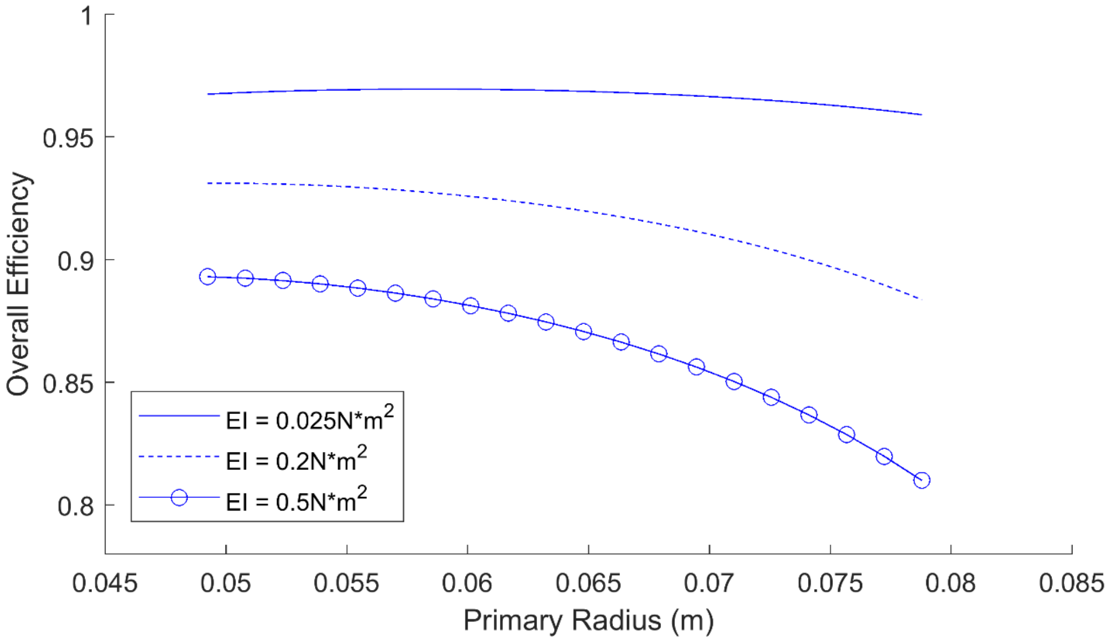


Figure 7.4. Plot of overall efficiency for different belt bending stiffnesses. Decreasing the bending stiffness shows a substantial improvement in efficiency, with the largest improvements occurring in high range.

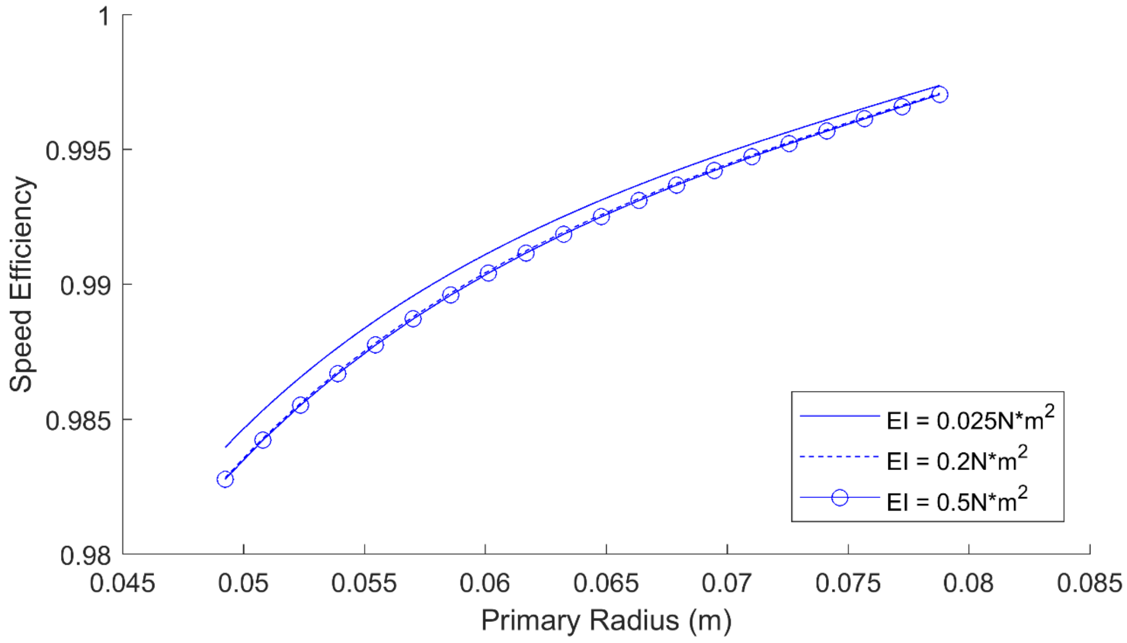


Figure 7.5. Plot of speed efficiency for different belt bending stiffnesses. Altering the stiffness of the belt does not show a significant change in efficiency.

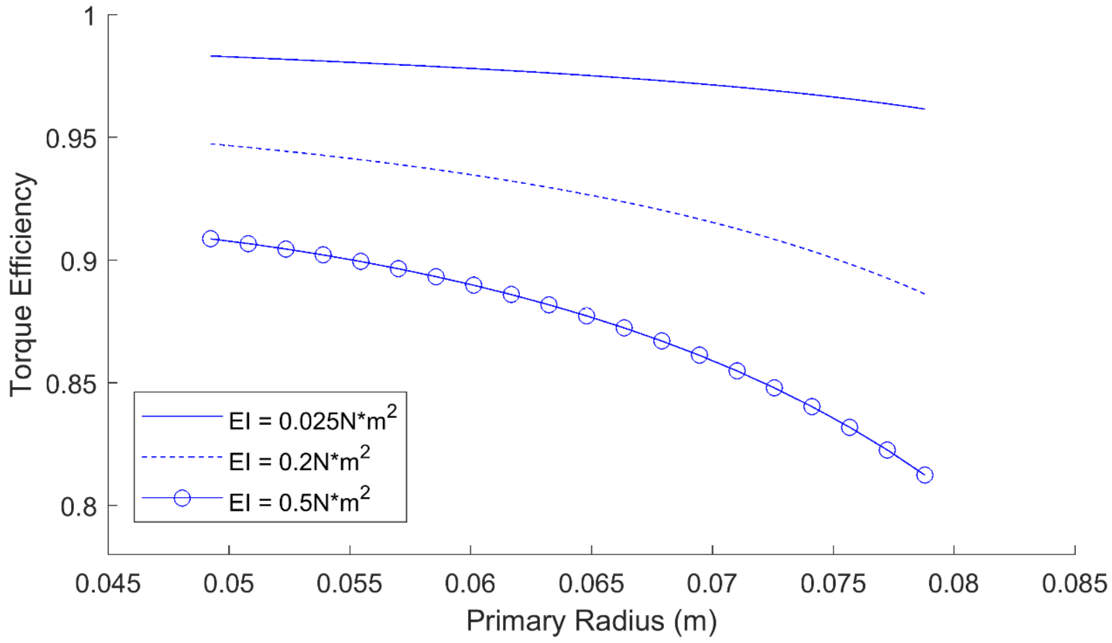


Figure 7.6. Plot of torque efficiency for different belt bending stiffnesses. Decreasing the bending stiffness shows the largest improvement in efficiency in high range.

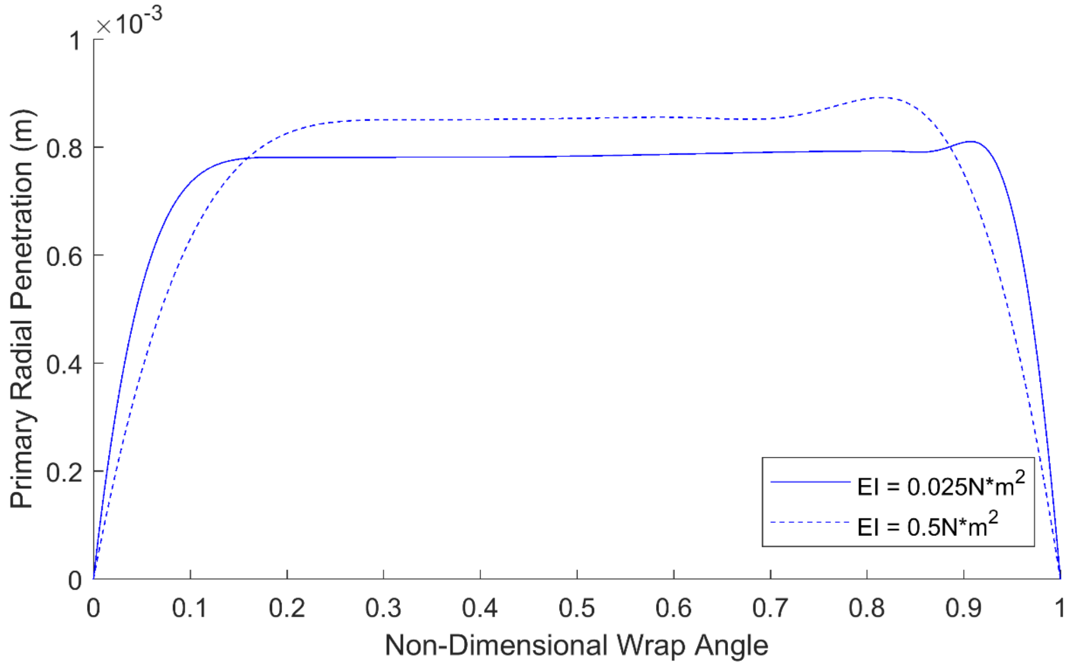


Figure 7.7. Radial penetration in the primary pulley by belt bending stiffness (low range). Increasing the bending stiffness results in longer entrance regions and higher radial penetrations.

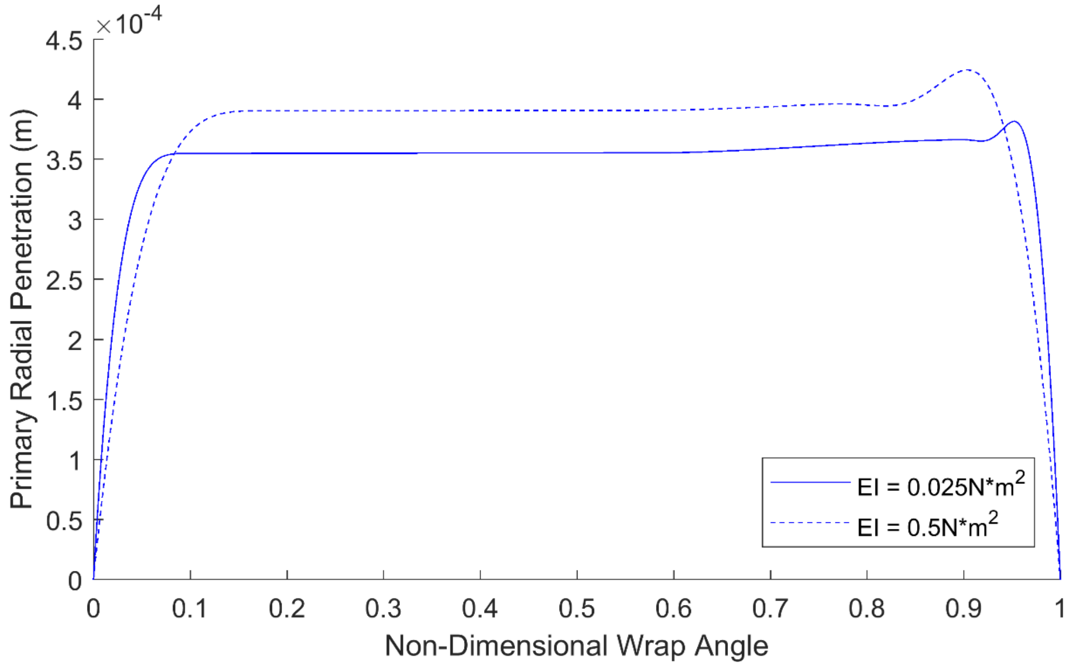


Figure 7.8. Radial penetration in the primary pulley by belt bending stiffness (high range).

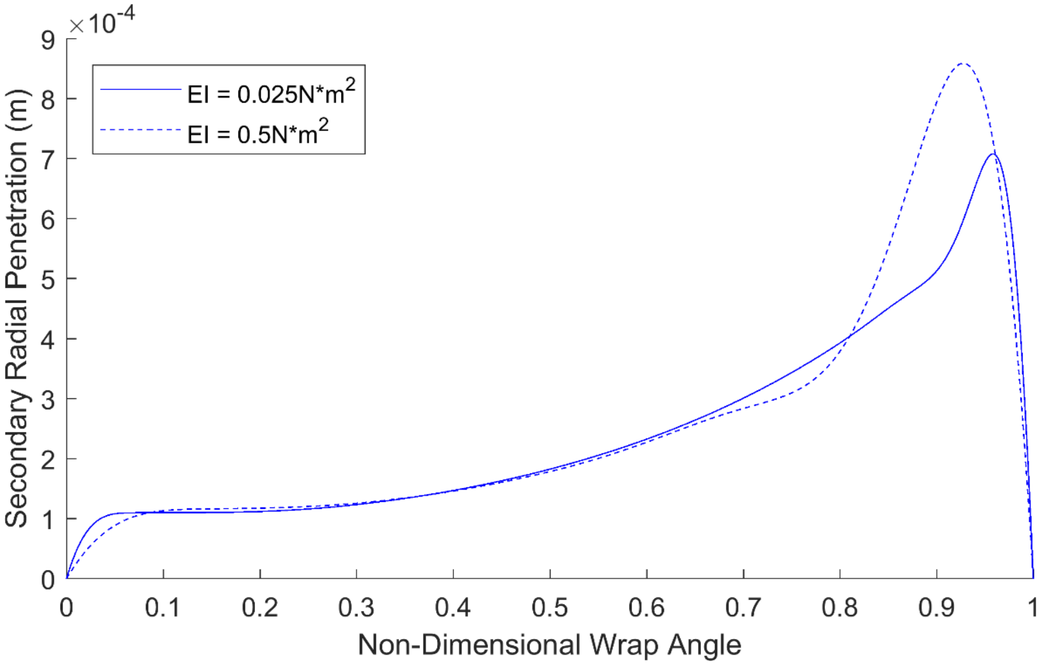


Figure 7.9. Radial penetration in the secondary pulley by belt bending stiffness (low range). The largest increases in seating are seen in the exit region.

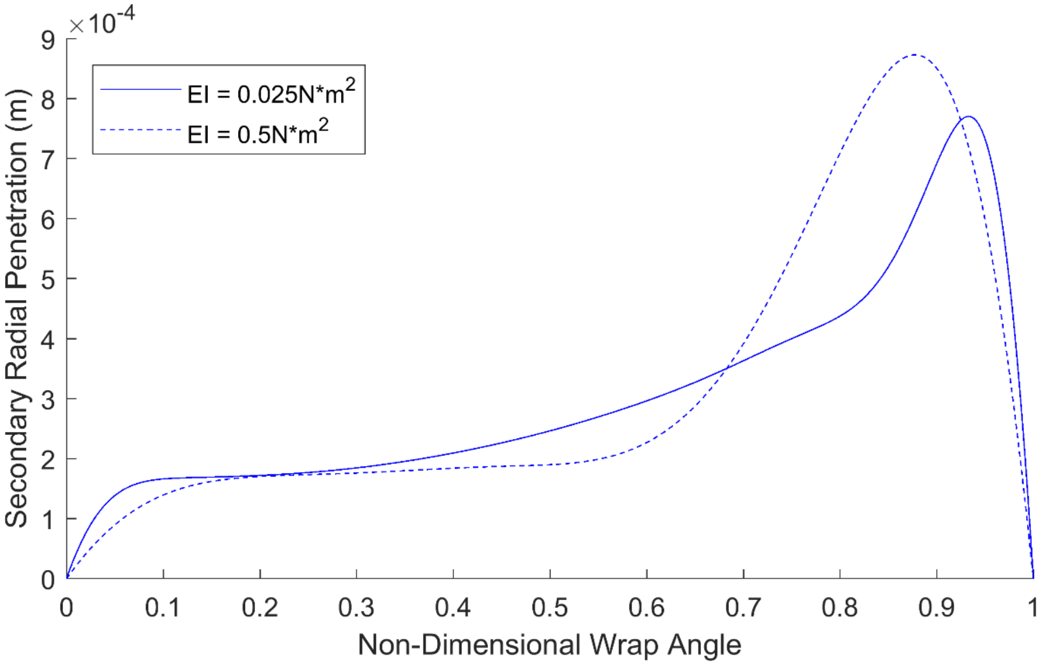


Figure 7.10. Radial penetration in the secondary pulley by belt bending stiffness (high range).

7.3 Slack Span Tractive Tension

As discussed in Section 5, the slack span belt tractive tension is chosen as a boundary condition because the tractive tension may never decrease below zero, providing a bound on the range of possible values. For a given belt drive, the driving torque and the radius may be used to approximate the difference in belt tensions across the spans. More information is needed to determine the values of the tensions, and in a CVT this information is contained in the axial force in the pulleys. Higher axial forces result in higher belt tensions. Plots of the efficiencies are shown in Figures 7.11 to 7.13. Plots of the axial forces in the primary and secondary pulleys for different slack span tractive tensions are shown in Figures 7.14 and 7.15, respectively.

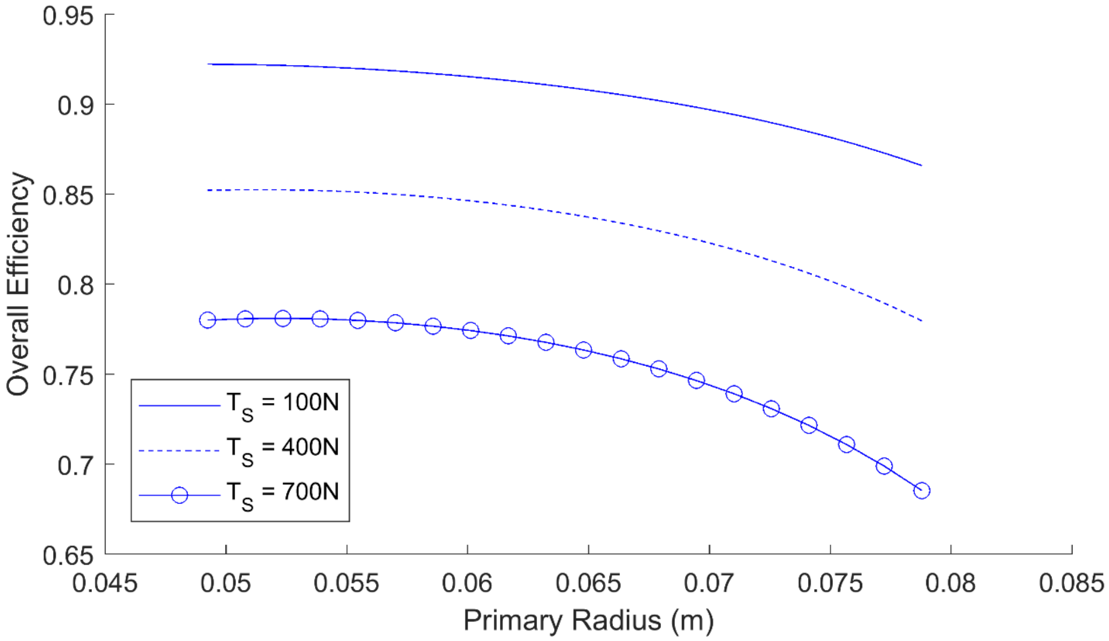


Figure 7.11. Plot of overall efficiency for different slack span tractive tensions. Increasing the slack span tension results in significant decreases in efficiency.

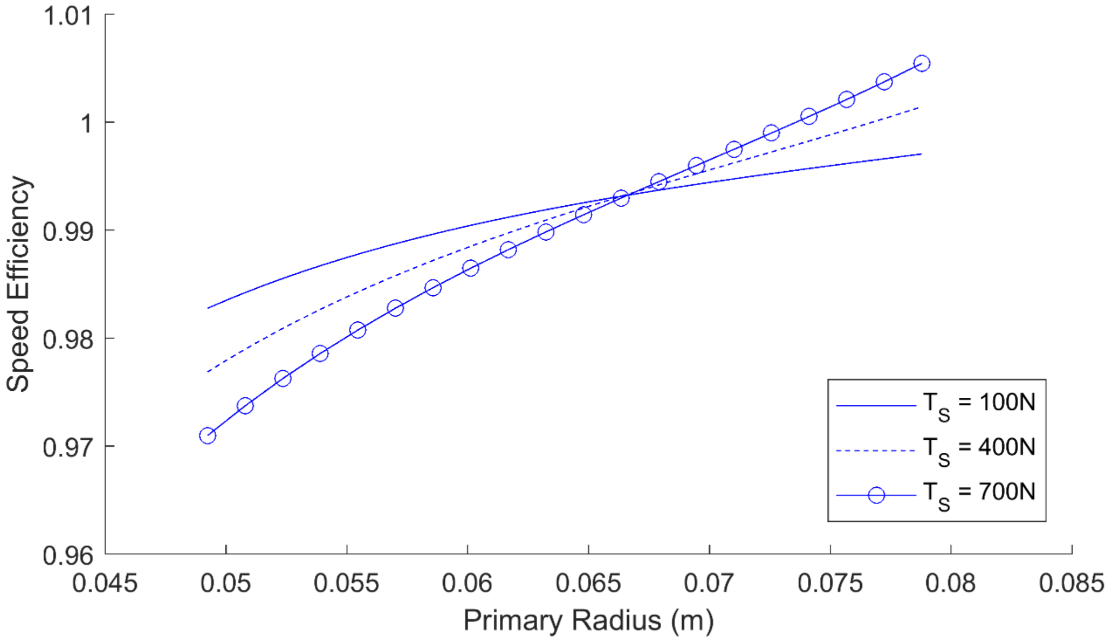


Figure 7.12. Plot of speed efficiency for different slack span tractive tensions. The data indicates that speed efficiency near a 1:1 ratio is independent of slack span tractive tension. Increasing the slack span tractive tension decreases the speed efficiency in low range, and increases the speed efficiency in high range.

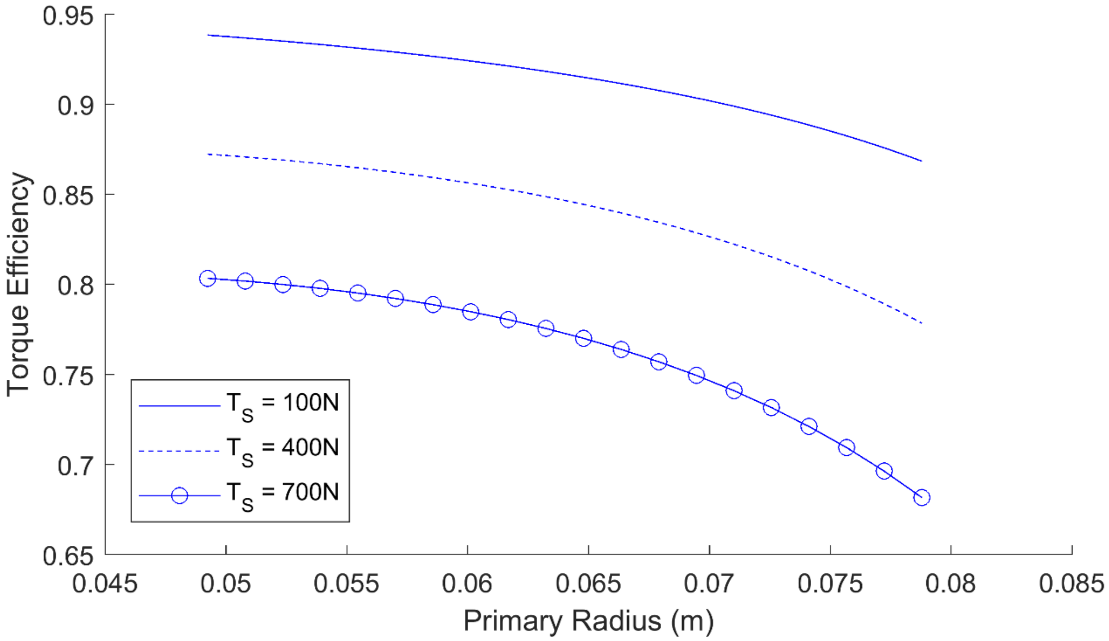


Figure 7.13. Plot of torque efficiency for different slack span tensions.

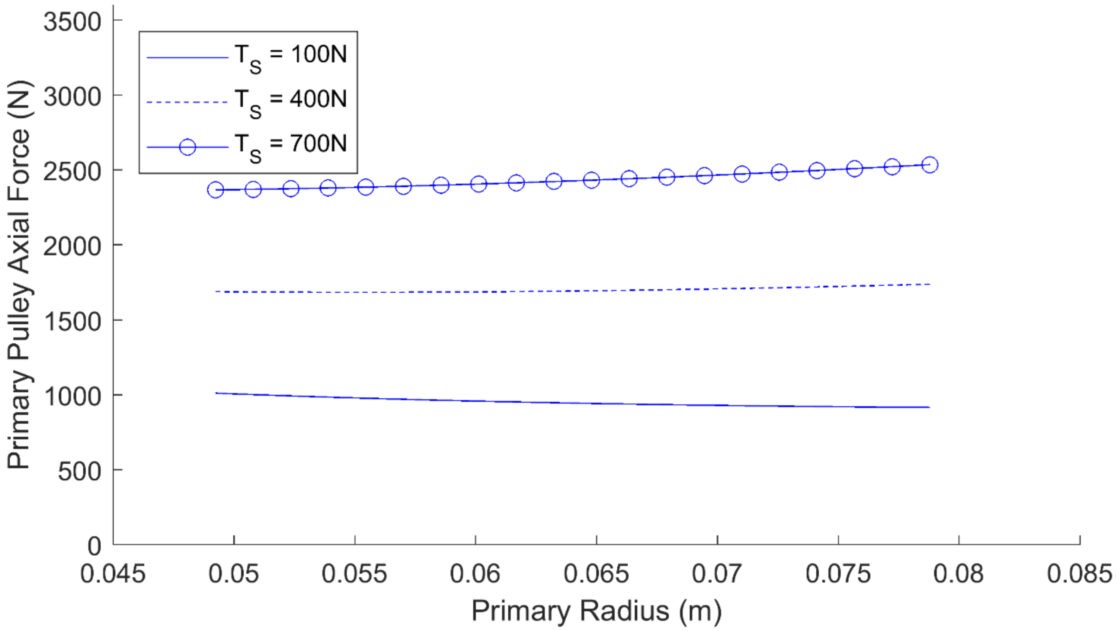


Figure 7.14. Plot of the axial force in the primary pulley by slack span tractive tension. The axial force appears to show a linear relationship with slack span tractive tension.

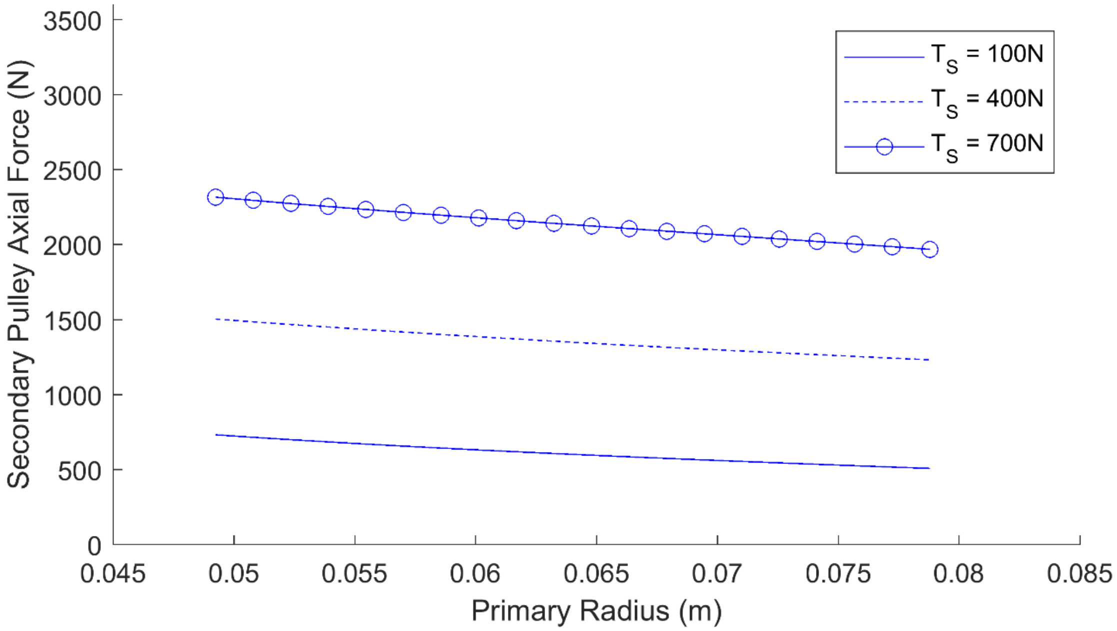


Figure 7.15. Plot of the axial force in the secondary pulley by slack span tractive tension. The axial force appears to show a linear relationship with slack span tractive tension.

7.4 Friction Coefficient

The coefficient of kinetic friction between the belt and the pulley sheaves may depend on several factors, such as temperature and sheave surface finish. It is not possible to determine this value experimentally and fully replicate the range of conditions that may be experienced during operation. Gerbert states that the friction coefficient is typically between 0.3 and 0.5, and that the specific value is of minor importance [7]. This assertion is evaluated here. Plots of the efficiencies for different values of the friction coefficient are shown in Figures 7.16 to 7.18. Changing the friction coefficient from 0.3 to 0.5 did not result in changes in overall efficiency over 1% at any point in the operating range, validating the use of 0.4 in this model. Errors in the friction coefficient do not propagate to other results of the model.

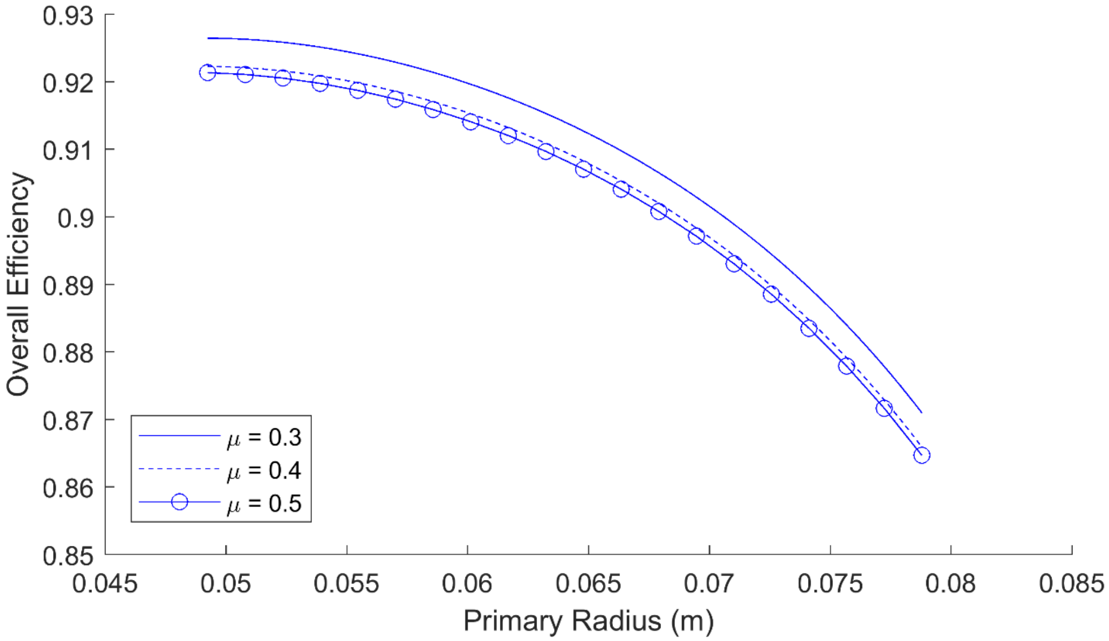


Figure 7.16. Plot of overall efficiency for different coefficients of kinetic friction. The efficiency does not show a strong dependence on the value of the friction coefficient.

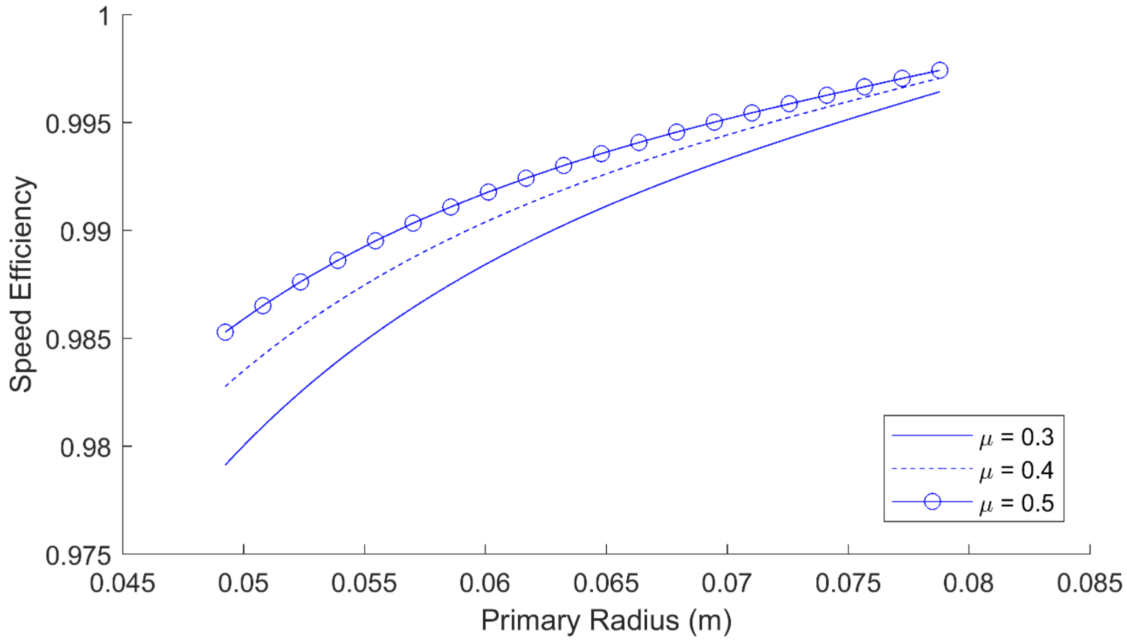


Figure 7.17. Plot of speed efficiency for different coefficients of kinetic friction.

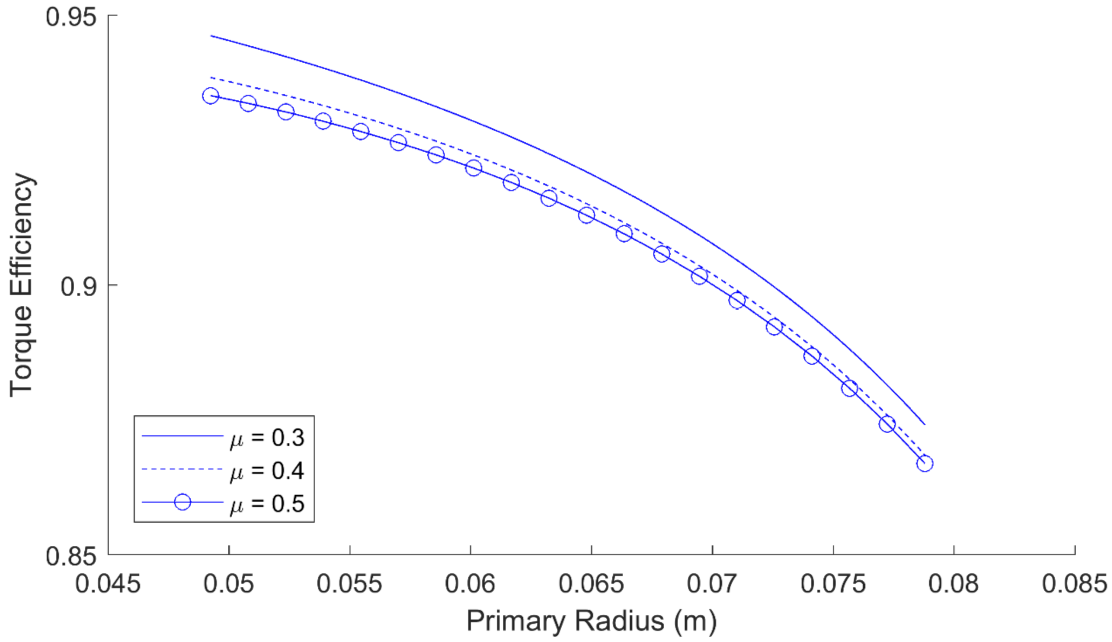


Figure 7.18. Plot of the torque efficiency for different coefficients of kinetic friction.

8 Design and Construction of a CVT Dynamometer

To validate the analytical belt model presented in this thesis, a CVT dynamometer was constructed. The dynamometer applies a torque load to the secondary pulley and measures the response. The sensors used are able to determine the efficiency of the CVT by comparing the output power data to known engine dynamometer data. Pictures of the dynamometer are shown in Figures 8.1 and 8.2.

To be viable for use with the Baja team, the test rig must be low-cost. For this reason, a data acquisition unit was designed that uses an Arduino to collect data, and a Raspberry Pi to log and process the data. The dynamometer steps the brake torque to control the operating point of the CVT and allows for the full automation of the test cycle.

8.1 Summary of Test Setup

To obtain useful data, the operating conditions of the CVT on the dynamometer must match the loading that is seen on a vehicle. The engine used for Baja SAE is a 10HP air cooled, 4-stroke, single cylinder design made by Briggs and Stratton. To provide the torque to the secondary pulley, an automotive friction brake is used. A screw-driven linear actuator is used to actuate the master cylinder of the braking system and control the brake pressure. A spring on the master cylinder pushrod decreases the sensitivity of the brake pressure to actuator position to allow for precise application of the torque. From the engine governor speed and the published maximum gear ratio of the CVTech CVT, the secondary pulley can reach speeds in excess of 7000RPM. Driving the brake rotor at this speed is not feasible, and a single stage gear reduction is used between the CVT output and the brake shaft at a 32:61 reduction. Due to the high efficiency of a gear train compared to that of a CVT, it is assumed that no power is lost in the reduction. The increased torque and reduced speed of the brake rotor shaft also allows for easier measurement of the load torque.

The secondary pulley is mounted on a ¾” keyed shaft, supported by bearings on both sides of the pulley. The engine is mounted to an aluminum base plate using four bolts. Slots for the bolts are milled into the base plate to allow for adjustment of the pulley center-to-center distance. The aluminum base plate is bolted to a welded steel frame using PVC vibration isolators. The steel frame may either be anchored into a concrete floor to reduce vibration, or used with casters to allow for it to be transported. A 0.060” thick steel safety cage encloses the CVT to prevent projectiles from injuring the operator in the event of a mechanical failure. The cage has a hinged lid that allows for easy access to the components, and is bolted together during operation.

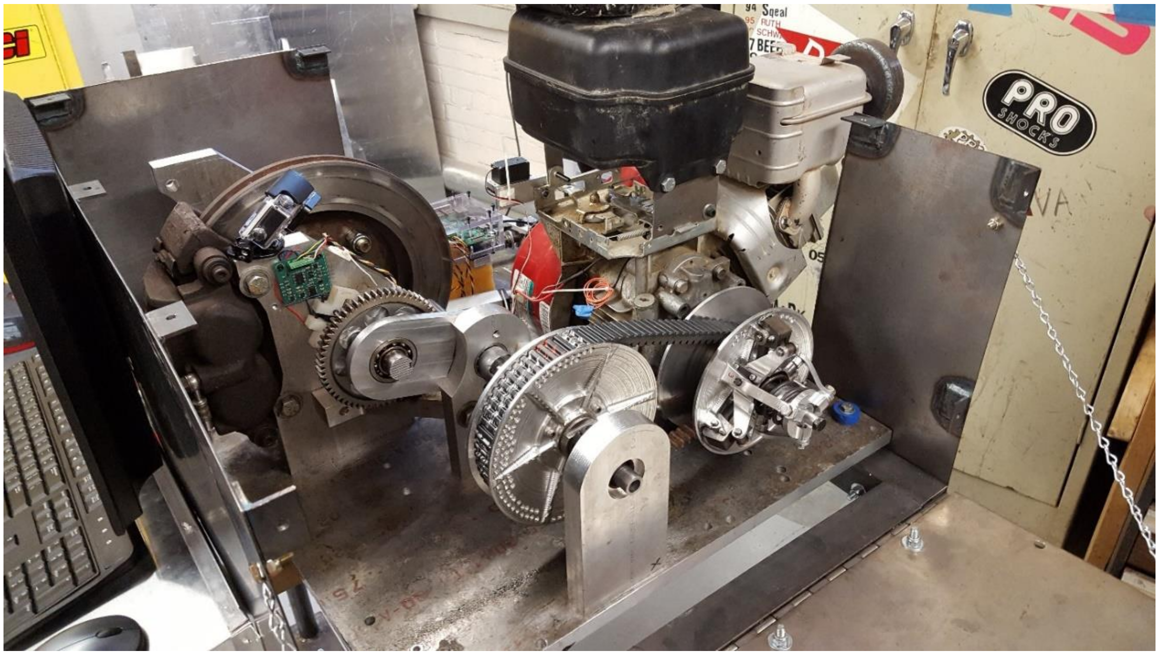


Figure 8.1. Inside of the CVT dynamometer. The custom CVT used with the 2018 team is shown in assembly.

A monitor and keyboard are mounted to the safety cage resulting in a fully self-contained assembly. This is shown in Figure 8.3. The monitor is used to display “live” data and provide a user interface during operation. It may also be used to modify the test code on the Raspberry Pi directly, without the need for an additional computer.

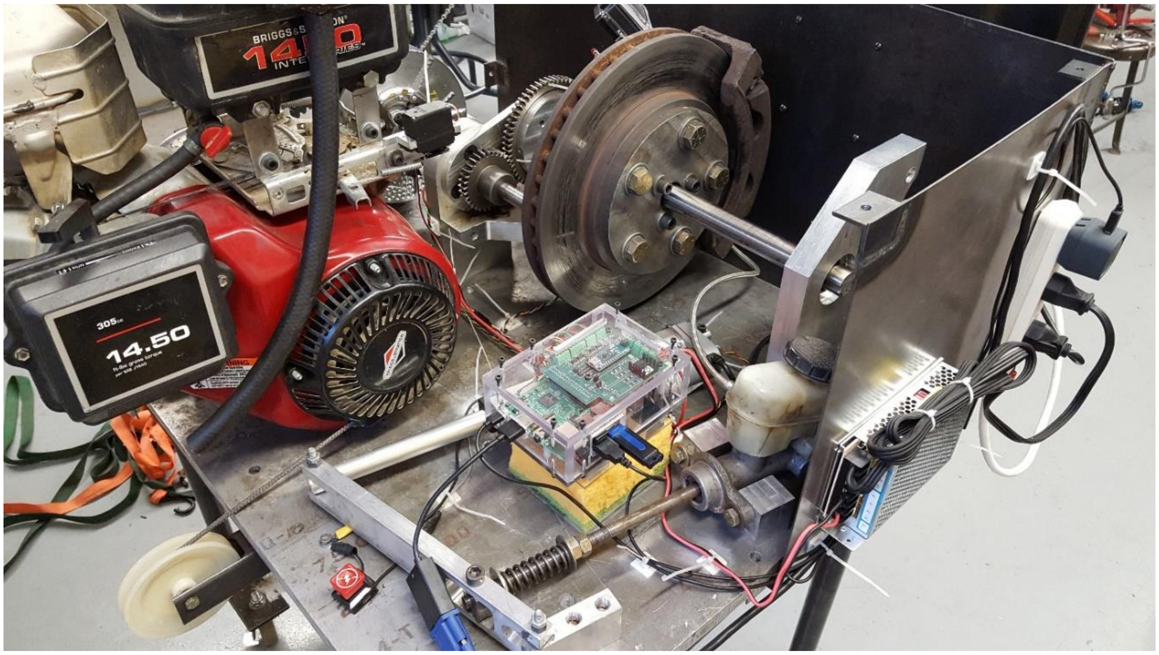


Figure 8.2. Back side of the CVT dynamometer. The friction brake and actuation system is shown, as well as the custom data acquisition unit mounted on a low-cost vibration isolator.

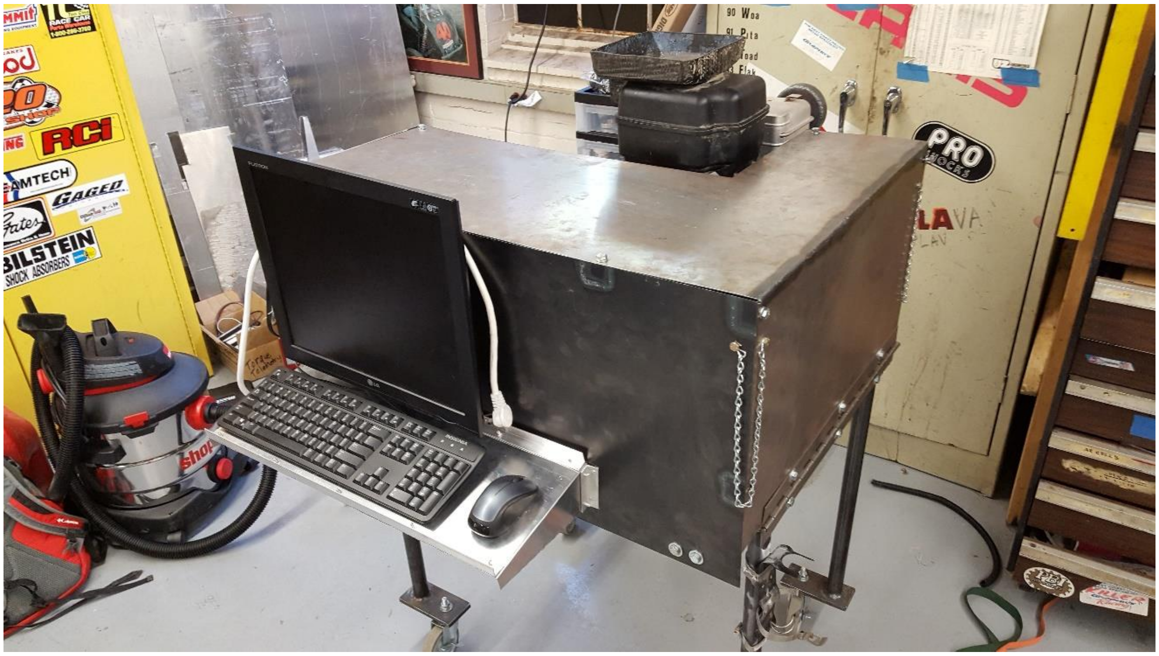


Figure 8.3. Dynamometer with the safety cage closed. An attached monitor and keyboard provide a user interface and allow for the system to be fully contained.

8.2 Data Acquisition System

Measurements of interest on the dynamometer are angular velocities for both pulleys, and the output torque of the transmission. The brake rotor temperature is also monitored to prevent overheating and damage during use. A low-cost data acquisition system is designed to take advantage of the fast handling of the GPIO's offered by an Arduino, and the processing power of a Raspberry Pi.

8.2.1 RPM Measurement

Both shaft speeds are measured using Melexis US5881 Hall Effect sensors [22]. These sensors are unipolar, and are only capable of detecting the south pole of a magnet. The sensor outputs a logic high in the absence of a magnetic field, and outputs a logic low when a field is incident on the sensor. The sensor contains a Schmitt trigger for cleaner switching and to add a hysteresis loop. The recommended circuit from the manufacturer is shown in Figure 8.4. The sensor is an open-drain device, meaning it is not capable of sourcing current, requiring a pullup resistor to bring the output to 5V when no magnetic field is present. A capacitor is used for hardware debouncing.

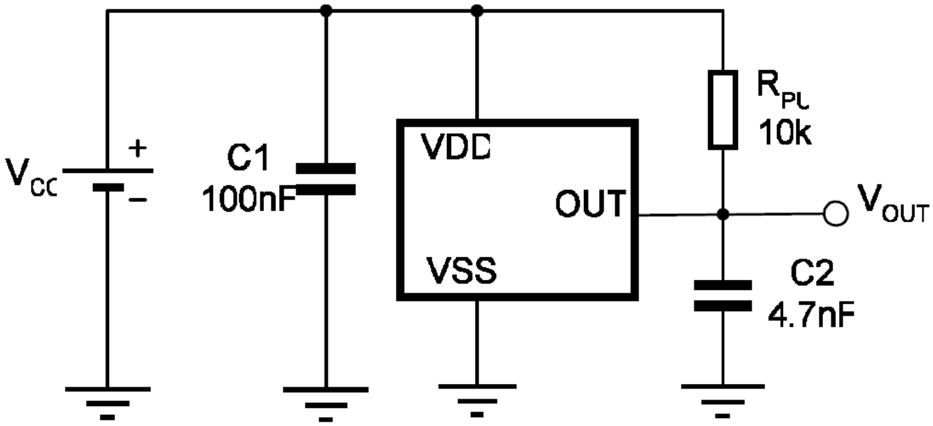


Figure 8.4. Recommended circuit for the Melexis US5881 Hall Effect sensor [22]. C1 provides smoothing for the supply voltage, and C2 is included for debouncing.

Pictures of the sensors are shown in Figures 8.5 and 8.6 for the primary and secondary pulleys, respectively. To reduce vibration and protect the sensors, both are embedded in machined nylon blocks and encased in epoxy. The sensor on the engine is bolted to the crankcase, and the sensor on the secondary pulley is bolted to the bearing support. The sensors are read using an Arduino Nano. The code for this is included in the Appendix. The Arduino Nano has two available interrupt pins. Coding the pins as interrupts is more efficient than a polling method, as the code does not execute until a hardware event occurs. Once a falling edge on the voltage signal is detected, the Arduino reads the current time from its onboard clock and calculates the speed in RPM using the time since the last trigger. The data that is saved by the Arduino is an unsigned 16 bit integer for the shaft RPM and an unsigned 32 bit integer for the timestamp in microseconds. The micros() function is used with the Arduino to return the timestamps, and it gives a value to a resolution of 4 microseconds. Since the sampling is determined by hardware events, it is nonuniform. Saving timestamps along with values allows the data to be resampled to a desired fixed rate during post processing if transient results are desired.

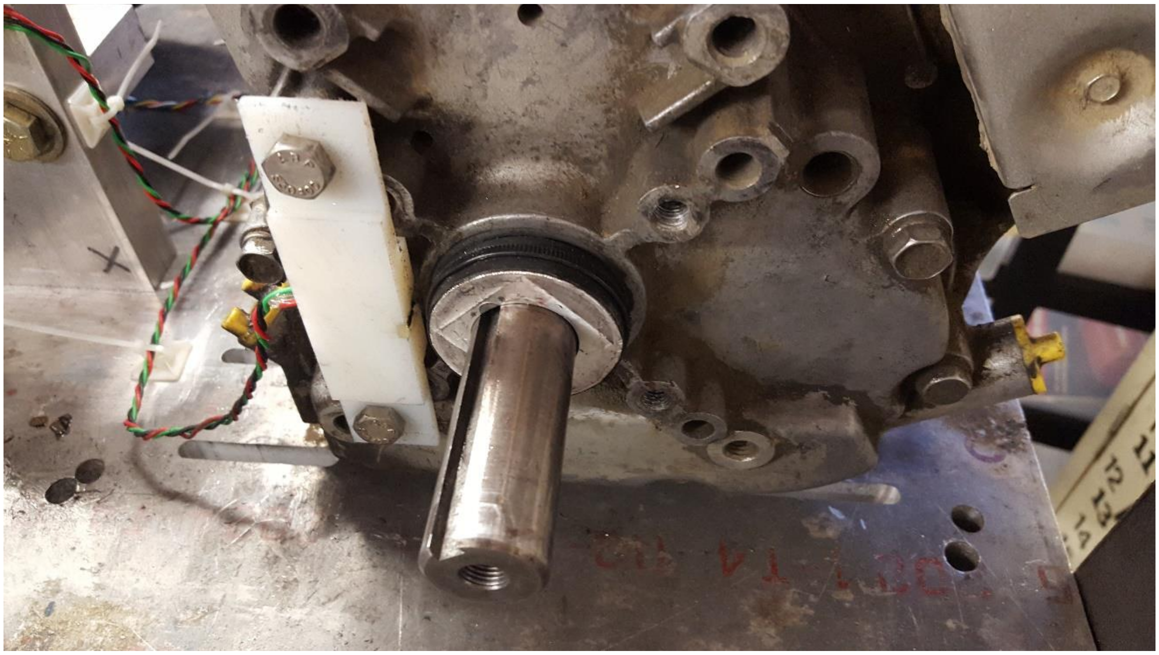


Figure 8.5. Hall Effect sensor used to measure the engine shaft speed. The sensor is embedded in a nylon block that is bolted to the crankcase of the engine. The sensor is encased in epoxy for protection.

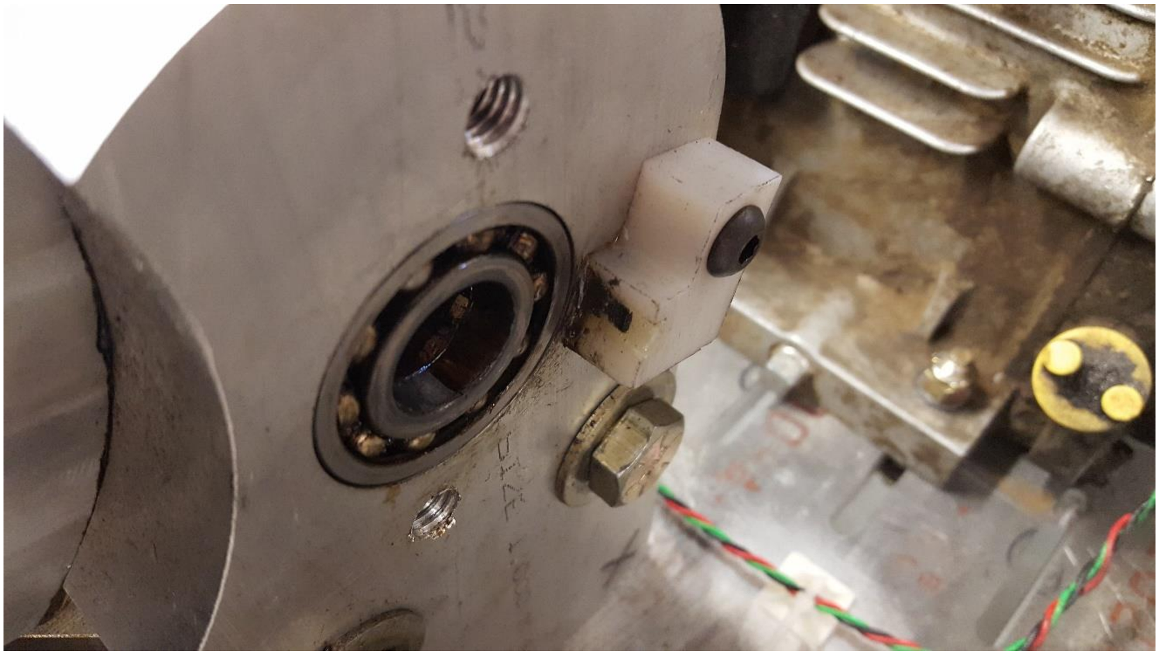


Figure 8.6. Hall effect sensor used to measure the secondary pulley speed. The sensor is bolted to the support next to the bearing for the secondary shaft.

Through trial and error, it is determined that the capacitor for hardware debouncing of the signal is not sufficient. The sensors often return two timestamps per rotation. To prevent the sensor from triggering twice, software debouncing is used in the Arduino code. The engine is governed to 3800RPM (period of 16ms), and the maximum expected speed for the secondary is approximately 7600RPM (period of 8ms). When an interrupt occurs, the current time is compared to the last time stamp. If the current time is less than 10ms after the previous time for the primary pulley, or less than 6ms for the secondary pulley, the event is ignored and no values are stored.

8.2.2 Torque Measurement

The output torque of the CVT is measured indirectly by measuring the torque on the brake rotor shaft, after the gear reduction. A commercially available rotary torque sensor is over the budget of the project and not a viable option. A lower cost alternative is to build a torque sensor using strain gages and a slip ring. To measure strain in a material, bonded foil strain gages may be adhered to the surface of the material under test, and will output a change in resistance proportional to the deformation in the material.

The percentage change in resistance is related to the strain in the material through a gage factor, defined as follows:

$$GF = \frac{\frac{\Delta R}{R}}{\varepsilon} \approx 2 \qquad (8.1)$$

This value is typically around 2 for a metallic strain gage [23]. To convert a change in resistance to a measurable voltage, a Wheatstone Bridge is used. A typical Wheatstone Bridge circuit is shown in Figure 8.7 and uses two voltage dividers to generate an output voltage, E_o. A full bridge configuration generates the largest possible output voltage for a given strain, and consists of four gages.

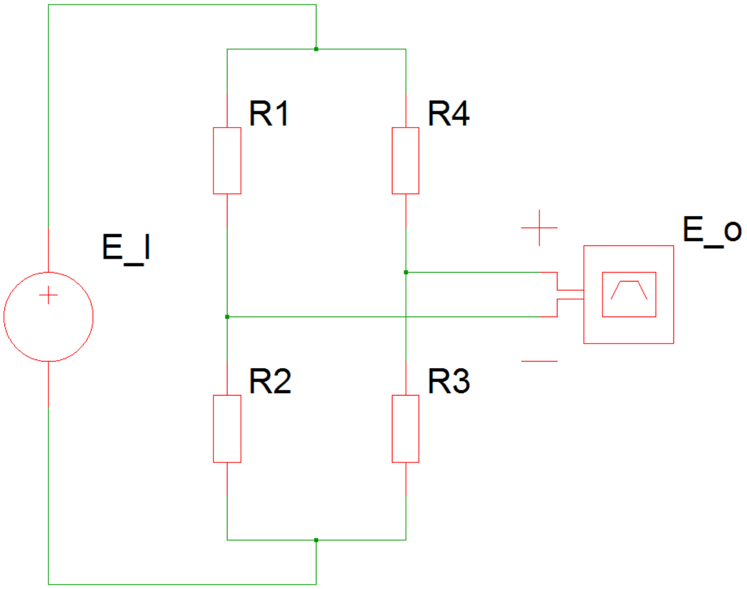


Figure 8.7. Typical Wheatstone Bridge application for strain gage circuits. The bridge excitation voltage results in a differential output voltage based on the value of the resistances. A full-bridge configuration consists of four gages, represented by R1 through R4.

A pure torsional load on a shaft results in two principal stresses, oriented 45° from the axis of the shaft, and the gages must be aligned with these principal stresses. Orienting the gages in this fashion offers temperature compensation, as well as compensation for residual axial and bending stresses [24]. This is illustrated in Figure 8.8.

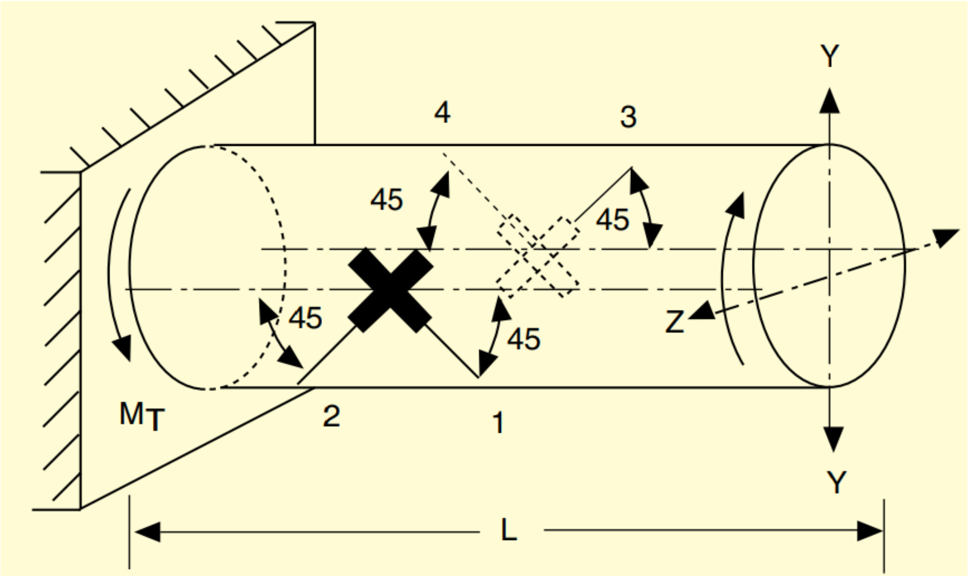


Figure 8.8. Strain gage orientation to measure torsional stresses [24]. Two gages are mounted on each side of the shaft at 45° angles to the axis of the shaft. This orientation provided compensation for temperature as well as axial and bending stresses.

Two strain gage rosettes are used, with each gage consisting of two elements, 90° apart, to create a half-bridge [25]. The bridge voltage is amplified using an AD8227 Instrumentation Amplifier, from Analog Devices. The amplifier reads the differential voltage from the bridge, multiplies it by a gain factor, and outputs a voltage with respect to a provided reference; ground in this case [26]. To find the output of the Wheatstone Bridge from a given shaft torque, the strains in the principal directions are calculated from the shaft elastic modulus and section properties:

$$\varepsilon = \frac{2\,T}{E\,\pi\,r^3}(1+\nu) \tag{8.2}$$

where T is the shaft torque, E is the tensile modulus, r is the shaft radius, and ν is Poisson's ratio. From the strain, the full bridge voltage is calculated using Equation 8.3:

$$E_o = E_i\,GF\,\varepsilon \tag{8.3}$$

The brake rotor shaft is a 1" diameter 1045 steel rod (E = 206GPa, ν = 0.35), keyed along the entire length. For simplicity, it is modeled as a solid 1" diameter cross section. The maximum expected torque this shaft will experience is approximately 100N*m, from the published engine torque, minimum

CVT ratio, and the gear reduction on the dynamometer. To account for possible transient effects, the gain is selected such that saturation occurs when over double the maximum expected torque is applied, with up to 1mV of unbalance present in the bridge. Using an excitation voltage of 5V, the maximum bridge voltage, with unbalance, is approximately 5mV. The gain resistor value in the AD8227 amplifier is calculated with Equation 8.4 [26]:

$$R_G = \frac{80k\Omega}{G-5} \tag{8.4}$$

A resistance of 113Ω corresponds to a gain of 713, and this results in a maximum expected amplified voltage of 3.74V. The amplifier, including the gain resistor and rail voltage smoothing capacitors, is mounted on a custom PCB that is fixed to the shaft. This is illustrated in Figure 8.9.

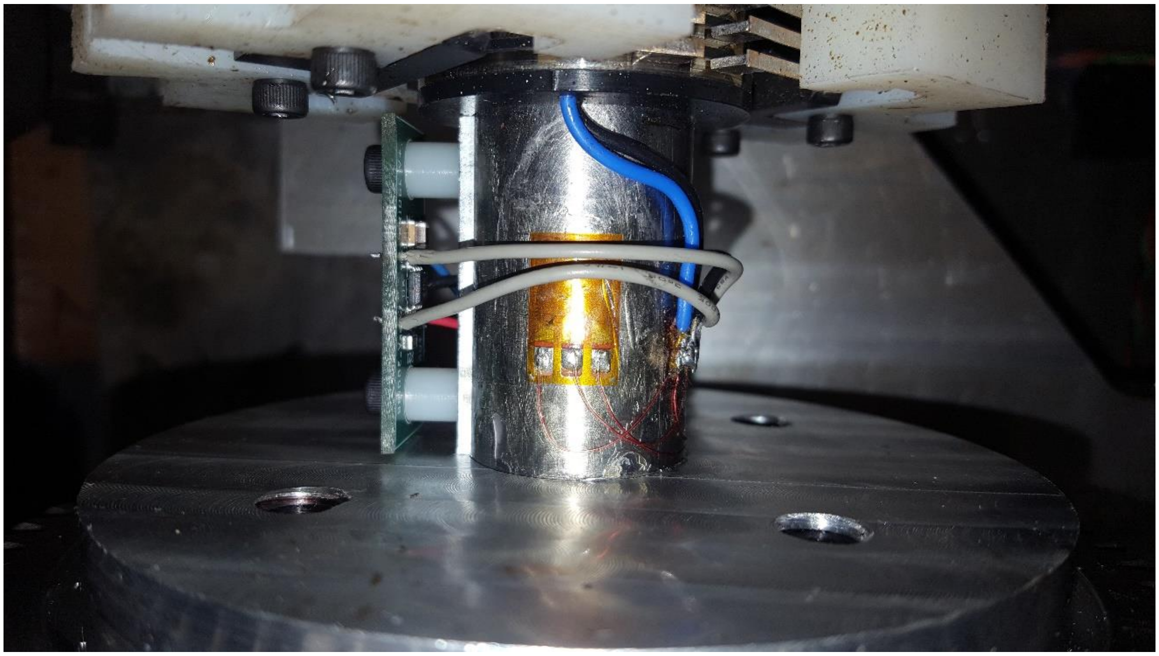


Figure 8.9. Strain gage rosette used to measure shaft torque. Also shown is the AD8227 instrumentation amplifier used to amplify the bridge voltage by a gain of 713.

To transmit signals to and from the shaft, a slip ring is used. Due to the large amounts of vibration present on the rig, two brush blocks are used, resulting in four brushes per ring. This is shown in Figure 8.10.

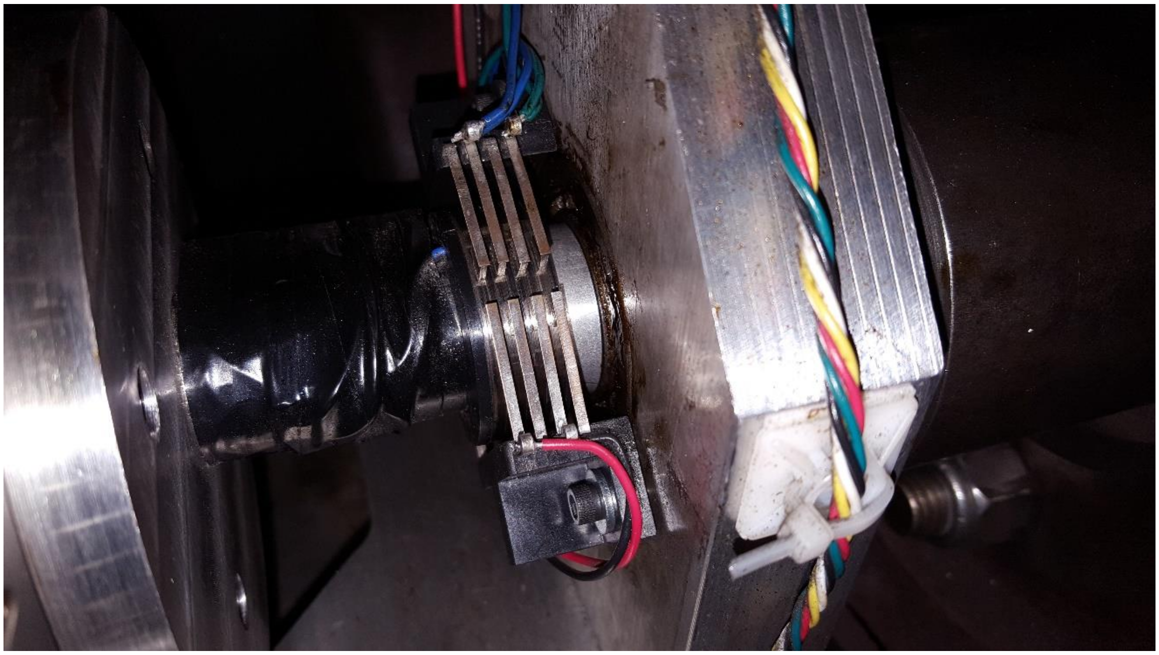


Figure 8.10. Slip rings used to transmit strain gage signals from the shaft.

To reduce the noise induced by the slip ring, a low cost lock-in amplifier is constructed in a manner similar to the one described in [27]. Lock-in amplifiers are used to remove noise from an AC signal, when the signal of interest is at a known frequency and phase. A demodulator is able to remove all components of a signal that are not at the same frequency and in phase with a reference signal, leaving only the signal of interest. This method is capable of recovering a signal from noise sources thousands of times larger in magnitude [28]. The AD630 modulator/demodulator is a low-cost option to accomplish this [29]. Since a dedicated function generator is not available for the dynamometer, an astable oscillator using a 555 timer is used to excite the bridge with a single-sided square wave instead, at approximately 850Hz. Using a square wave as the reference signal does not offer as good noise rejection as a sine wave, as higher harmonics are present in the signal. The demodulator is only able to reject noise at frequencies other than the carrier wave frequency, and the use of a square wave allows noise at the frequency of the higher harmonics contained in the square wave to pass through. Since the expected noise sources are at frequencies much lower than the carrier wave frequency due to the relatively low shaft speeds, a square wave signal is sufficient in this application.

The output of the demodulator is a rectified square wave, with the same peak to peak amplitude as the original signal when configured to a gain of 1. This signal is then passed through a low pass filter to return a DC average that can be read by an analog-to-digital converter. The low pass filter is designed using the Filter Wizard from Analog Devices and the resulting schematic is shown in Figure 8.11 [30]. The frequency response of the filter is shown in Figure 8.12. The cutoff frequency is 15Hz at -3dB attenuation, and the stopband is 100Hz, at -40dB attenuation. The actual circuit used, with common resistor and capacitor values, is shown in Figure 8.13.

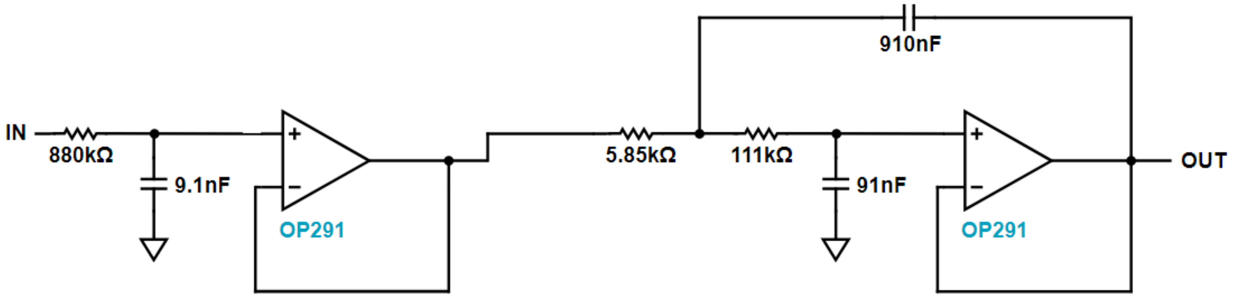


Figure 8.11. Active low pass filter design created with the Analog Devices Filter wizard [30]. The filter contains two stages, using a first-order buffered RC stage and a second-order Sallen Key stage.

The analog-to-digital converter used to read the DC output of the filter is an AD7171, 16 bit sigma-delta converter. The ADC samples continuously at 125Hz, and passes the value to the Arduino using two-wire serial communication [31]. Similar to the RPM measurement, the Arduino saves a timestamp along with each measurement from the ADC. The ADC sampling is unsynchronized with the RPM measurements, and these timestamps will allow all data to be resampled to the same uniform rate during post-processing if transient data is desired. A picture of the custom PCB containing the circuit in Figure 8.13 and the ADC is shown in Figure 8.14.

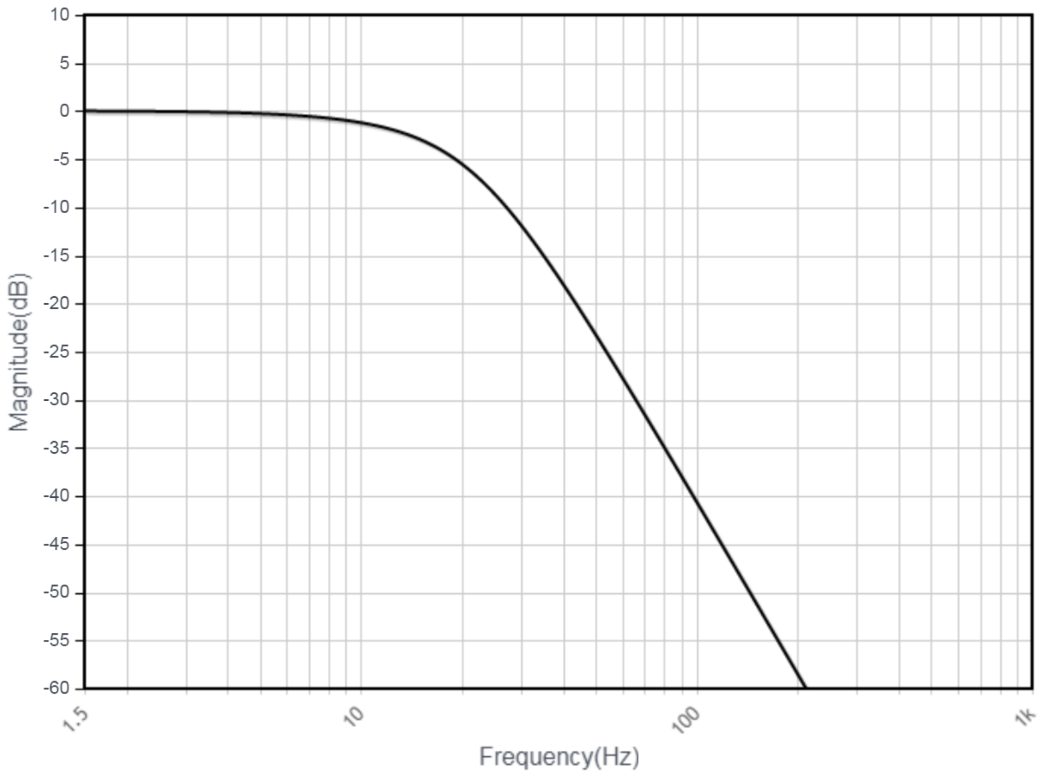


Figure 8.12. Bode plot for the filter used to find the DC average of the demodulator output [30].

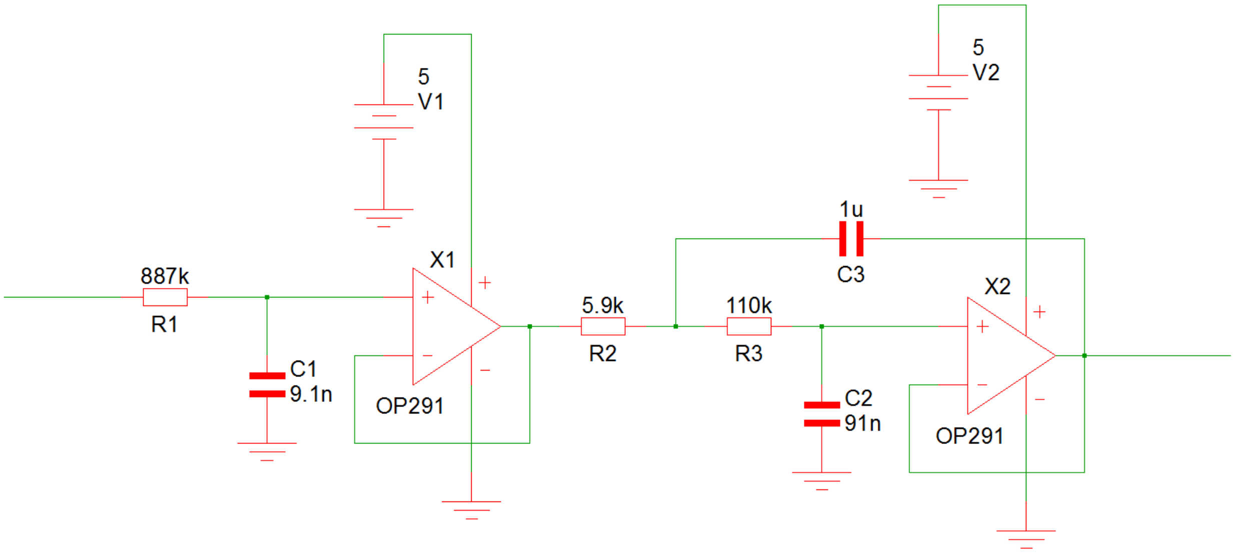


Figure 8.13. Filter schematic used in the final design, with common resistor and capacitor values.

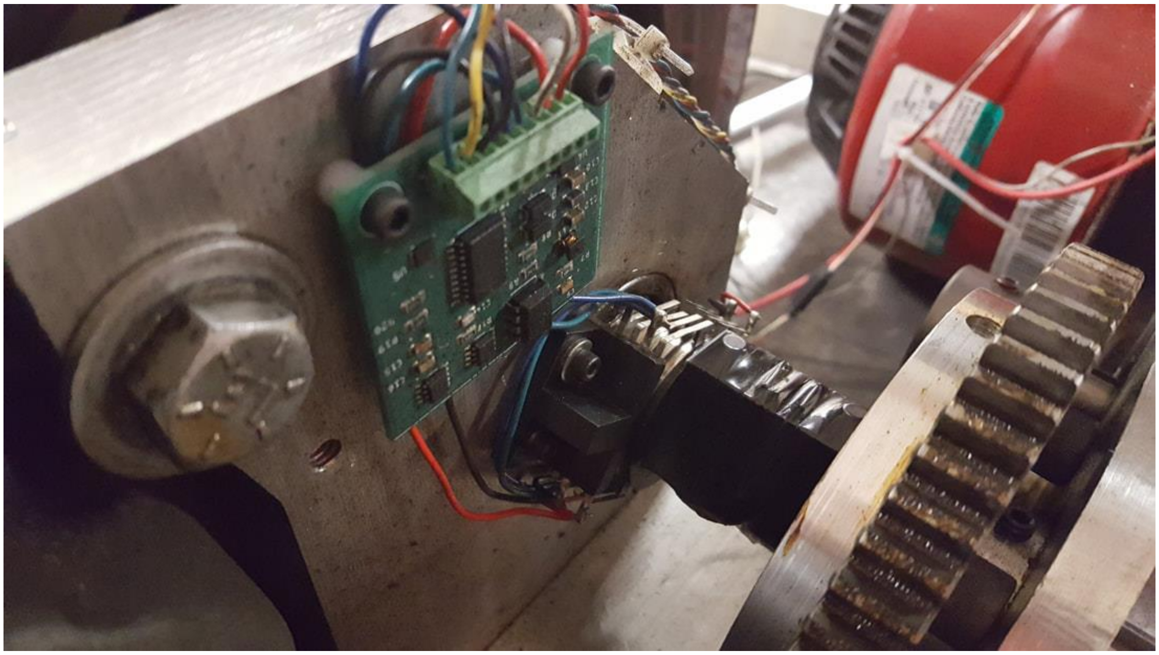


Figure 8.14. Custom PCB containing the lock-in amplifier and ADC. This board generates a square wave to excite the strain gage bridge, demodulates the signal, and filters the output to generate a DC voltage that is read by a 16 bit ADC.

8.2.3 Raspberry Pi/Arduino based Data Collection

An Arduino Nano is a prototyping platform based around the Atmel ATmega328P 8-bit microcontroller. The device contains 22 GPIO pins, 14 of which are digital and 8 are analog with built in 10-bit ADC’s. The small form factor of the Arduino Nano makes it suitable for use with a breadboard, or for it to be integrated into a larger system [32]. A Raspberry Pi is a general purpose computer built around a Broadcom System on a Chip (SoC) that is capable of running a Linux-based operating system [33]. Although the clock speed of the Raspberry Pi is substantially higher than that of the Arduino (1.2GHz versus 16MHz), it is unsuited to real-time applications due to the large overhead associated with the Linux operating system [33]. In contrast, the Arduino is a real-time device, as it runs compiled code that will take a fixed number of clock cycles to execute a task. For this reason, the Arduino is used for handling the time sensitive tasks on the dynamometer, including measuring the time between revolutions and reading the values from the ADC for torque data.

The RPM and torque data are collected by an Arduino, then passed to a Raspberry Pi for logging and processing using I^{2}C, a form of two-wire serial communication. To ensure that an error is not present in the transmission between the two devices, a simple checksum is used. Additional bits are appended to the end of the transmission, representing the sum of the data bytes sent prior. When the Raspberry Pi receives the data, it recalculates the sum of the data bytes and compares it to the checksum. If the values match, it is assumed that the transmission is successful and the data point is recorded.

Another custom PCB contains the supporting circuitry for the Hall Effect sensors and provides a mounting point for the Arduino. In addition, it contains a logic level translator to convert the 3.3V I^{2}C signal from the Raspberry Pi to a 5V signal for the Arduino, as well as the circuitry used for automation. The brake linear actuator direction is determined using an H-bridge made with a DPDT relay, and a MOSFET is used to turn the actuator on and off. An additional relay is connected to the engine kill switch to give the system the ability to shut down the engine remotely in the event of a problem, or when the test is completed. All sensors are connected to the board using screw terminals. A picture of the complete Raspberry Pi/Arduino system is shown in Figure 8.15.

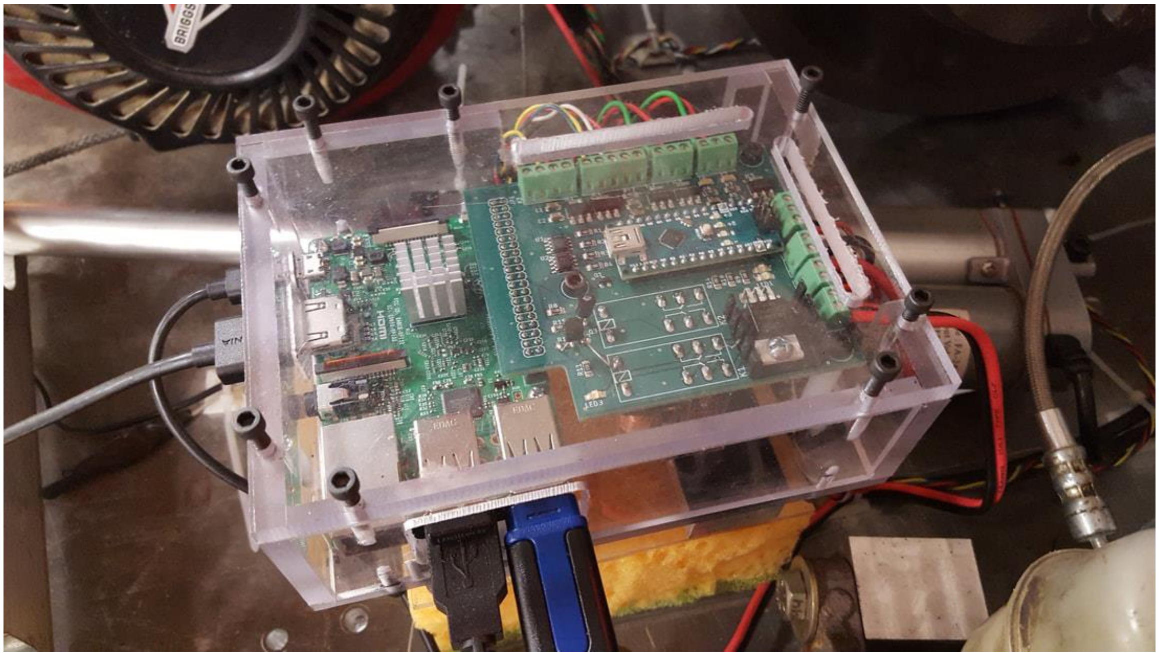


Figure 8.15. Custom Raspberry Pi/Arduino based data acquisition unit. The PCB containing the Arduino and all supporting circuitry plugs directly into the GPIO header on the Raspberry Pi.

9 Experimental Validation of Efficiency Results

To validate the analytical results, the CVT must be tested at steady-state. To ensure that the CVT is not shifting while data points are being taken, the brake load is incremented and the system is given time to settle. An ideal CVT would shift as necessary to result in a constant engine operating point, regardless of vehicle speed. Even with tuning, this is not possible with the stock CVTech CVT, and the engine speed varies with secondary RPM. Due to the relationship between speed and torque for the engine, this results in a changing primary pulley torque boundary condition as the CVT moves through its operating range. In addition, the slack span tractive tension is not known. Pulley axial force may be used as a boundary condition to solve for this. To make a valid comparison between the analytical and experimental results, the engine operating point boundary conditions and parameters must be adjusted in the analytical model before calculating efficiency.

9.1 Summary of Test Method

In theory, it is possible to design a feedback loop to apply the brake in order to control the operating point of the CVT. This is difficult in practice for several reasons. Brake torque is one of the possible outputs for a feedback loop. If the friction coefficient between the brake pads and rotor does not change and the rotor speed is not changing, the torque has a linear relationship to actuator position from the spring on the master cylinder pushrod. In practice, these assumptions do not hold and the engine noise included in the measurement makes this unsuitable for use in a feedback loop. Another possible output is the secondary pulley speed. For a constant power input, torque and speed on the pulley have an inverse relationship. This results in a non-linear plant model for speed vs actuator position and makes the determination of gains difficult. In addition, there is a significant time delay between a change in torque and a response in the CVT for the model tested. For these reasons, a feedback loop is not used for steady-state measurements.

To ensure a distribution of data points within the operating range of the CVT, the brake actuator is applied in small increments to gradually step the brake torque applied to the secondary pulley. After the load is increased, the system is given time to settle. Experimental results indicate that four seconds are sufficient to reach steady-state. Data is then collected for a period of 3 seconds before stepping the brake and moving to the next operating point. The Hall Effect sensors and the ADC do not sample at the same rates, with the highest sampling rate coming from the ADC at 125Hz. To ensure that all values from the Arduino are collected, the Raspberry Pi oversamples the Arduino at a rate of 150Hz. This results in duplicate values for many readings that are removed during post-processing. Since engine noise due to firing is not a parameter of interest in the analytical model, it must be filtered in order for comparisons to be made. This is accomplished by averaging all of the data points for speed and torque that are collected during the three second sampling time to result in a single data point, corresponding to an operating point for the CVT.

This process continues until the secondary pulley speed decreases below a threshold value. A user interface updates every second to display the most recent measurements, as well as monitor the temperature of the brake rotor. The user interface also has buttons for the user to begin the test once the engine is started, and an emergency stop in the event of a mechanical failure or excessive brake temperatures. Python code for the steady-state test method is included in the Appendix.

9.2 Torque Sensor Calibration

The strain gage measurement system must be calibrated to account for any possible bridge imbalance and stiffness effects due to the keyway, which is not accounted for in the gain calculations. To apply a known torque to the brake, the secondary pulley is fixed, a rope is wrapped around the brake rotor, and known weights are added to a bucket. The size of the weights added gradually increases to show the sensitivity of the sensor for small applied torques. A plot of the calibration curve is shown in Figure 9.1. Torque for the secondary pulley is calculated from the gear ratio. For each applied torque, 500 data points are collected from the ADC to show the repeatability of the measurement. The raw output of the ADC and the torque at the secondary pulley in N*m are related through the following equation with an R^2 value of 0.9997:

$$ADC = 108.19\,T + 35145 \tag{9.1}$$

The values for slope and intercept for each calibration run are saved in a .csv file on the Raspberry Pi, and the most recent values are imported each time a test is run.

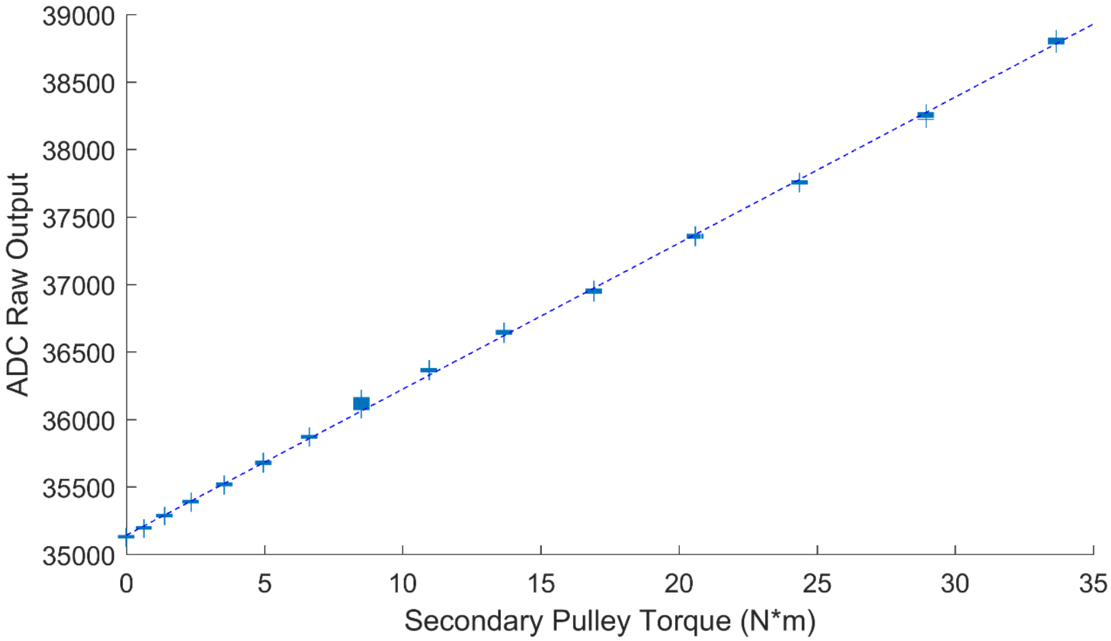


Figure 9.1. Calibration curve for torque data. 500 data points are collected for each applied torque. A linear fit matches the data well with an R^2 value of 0.9997.

9.3 Efficiency Results for a Commercially-Available CVT

In order to calculate the efficiency of the transmission, the engine power must be known. Due to the setup on the CVT dynamometer, the only way to determine engine power is by replacing the CVT with a chain drive, or other means of power transmission. This would introduce additional losses in the system and skew the results. A member of the University of Arkansas Baja team provides engine dynamometer data online in the Baja SAE forums for both new and used engines [34]. This data is obtained using a dynamometer typically used for go-kart engines. A data vector is extracted from this plot using the GRABIT function from the MathWorks File Exchange [35]. To determine the engine power at each corresponding data point, the engine RPM is used to interpolate the engine power from the data vector. This is compared to the measured output power to determine efficiency.

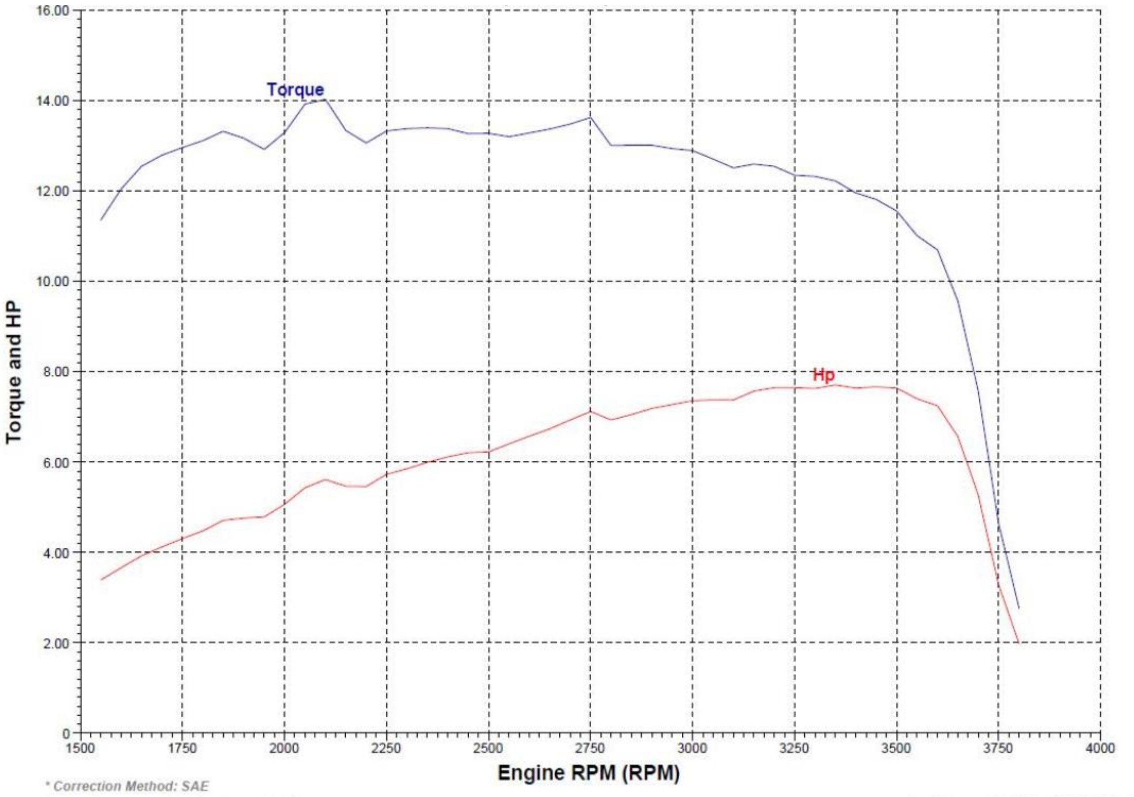


Figure 9.2. Dynamometer data for torque and horsepower output for a used Baja SAE engine [34].

Three different data runs were conducted, and the results are shown in Figures 9.3 to 9.6. All results are plotted against the measured secondary RPM. For these tests, the engine governor is set to the Baja SAE regulated value of 3800RPM. A typical data set for the torque, recorded over three seconds, is shown in Figure 9.7. This indicates the magnitude of the engine noise present in the system, even with a low pass analog filter used on the signal. An FFT of the signal with the mean removed is shown in Figure 9.8 to indicate the sources of noise present in the data. To account for the ADC clock inaccuracies and missed data points due to transmission errors, the data is resampled to 125Hz. The largest magnitude frequency components are due to the engine firing and the brake rotor shaft frequency.

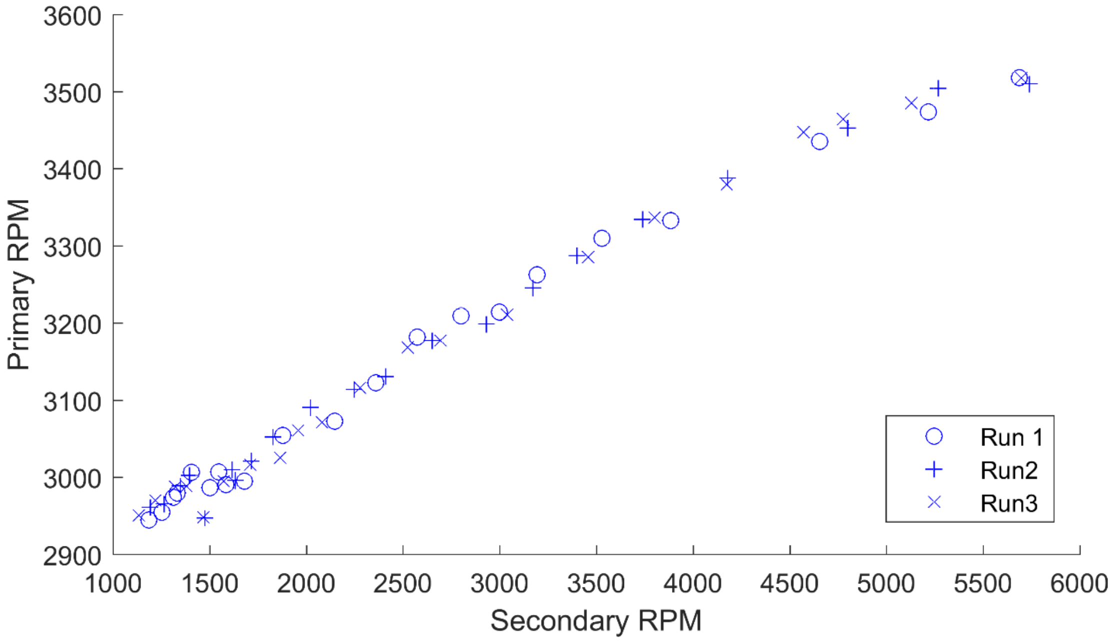


Figure 9.3. Experimental results for primary RPM vs secondary RPM. These results indicate the quality of the CVT tuning, as the primary RPM should remain constant as the secondary RPM changes.

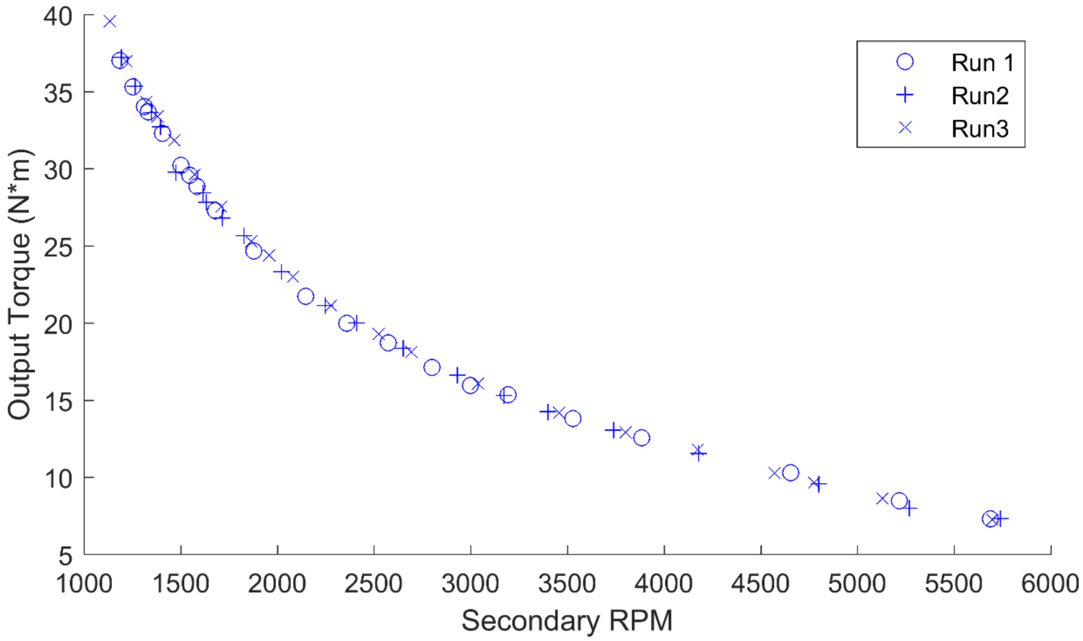


Figure 9.4. Experimental results for output torque vs secondary RPM. For a constant input power, this should follow a hyperbolic relationship.

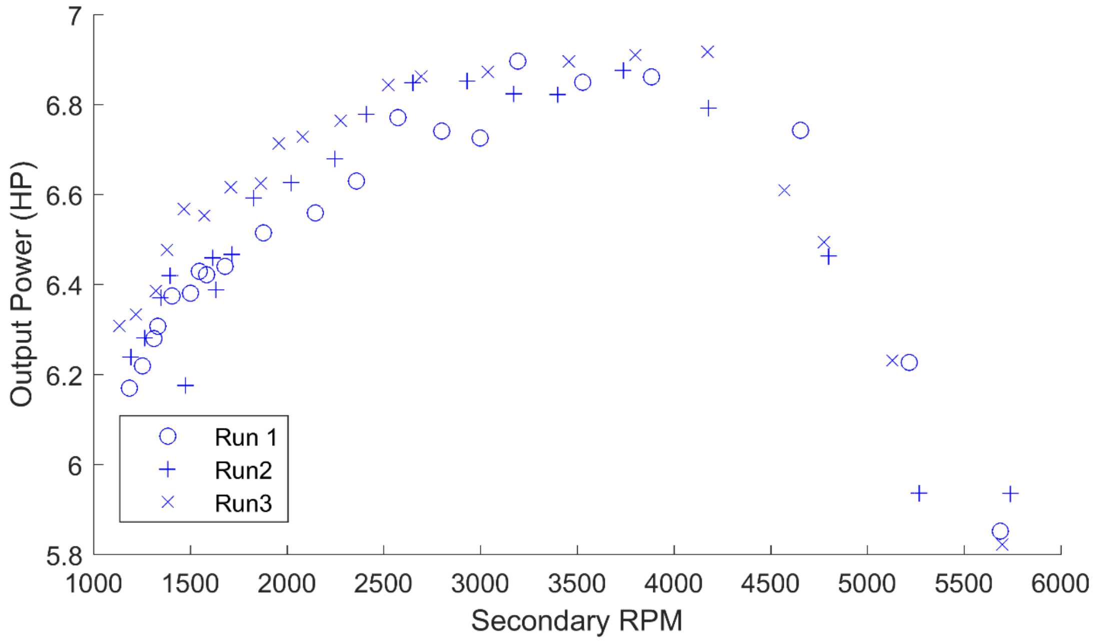


Figure 9.5. Experimental results for output power vs secondary RPM. The highest power output occurs for a secondary speed around 3500RPM, and this value decreases more sharply toward high range than low range.

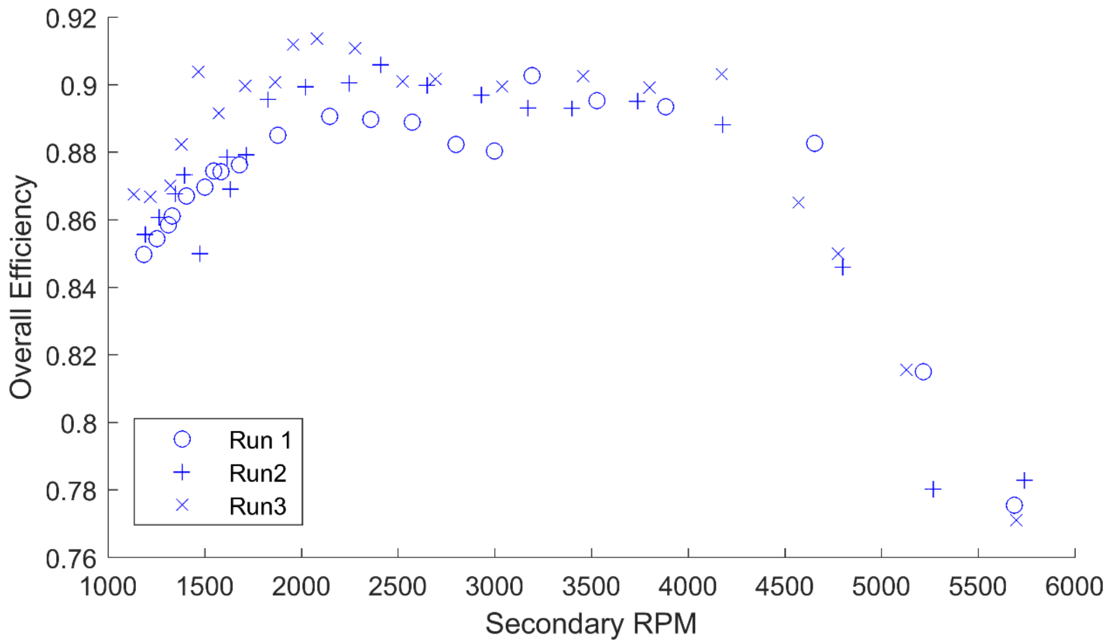


Figure 9.6. Experimental results for overall efficiency vs secondary RPM. The highest efficiency occurs at a secondary speed of approximately 2000-2500RPM and decreases more sharply toward high range than low range.

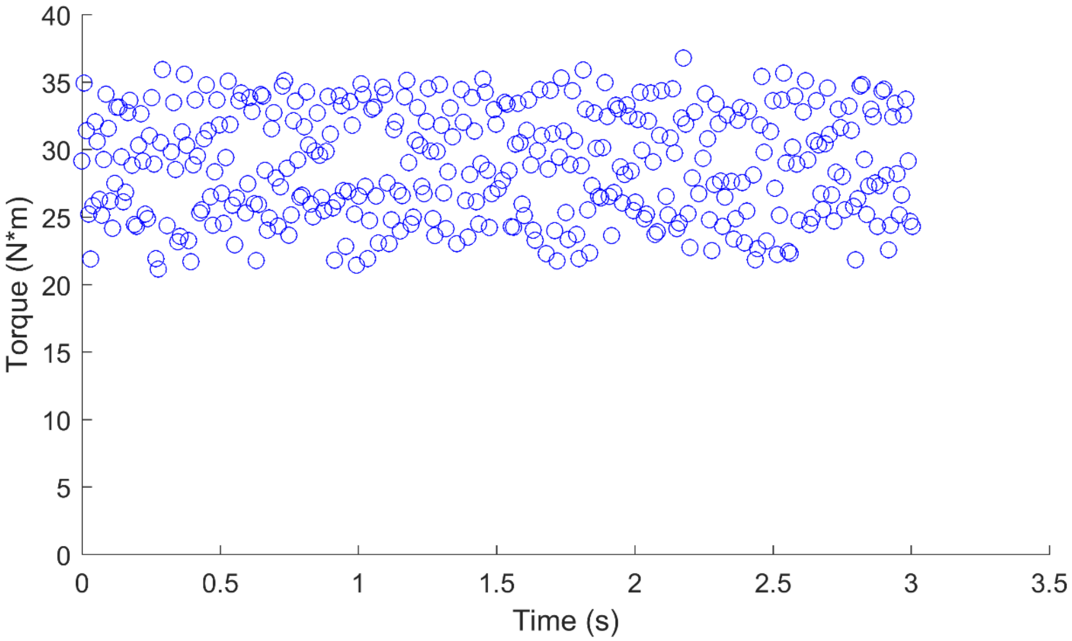


Figure 9.7. Plot of typical torque vs time data. This data corresponds to a secondary speed of approximately 1585RPM.

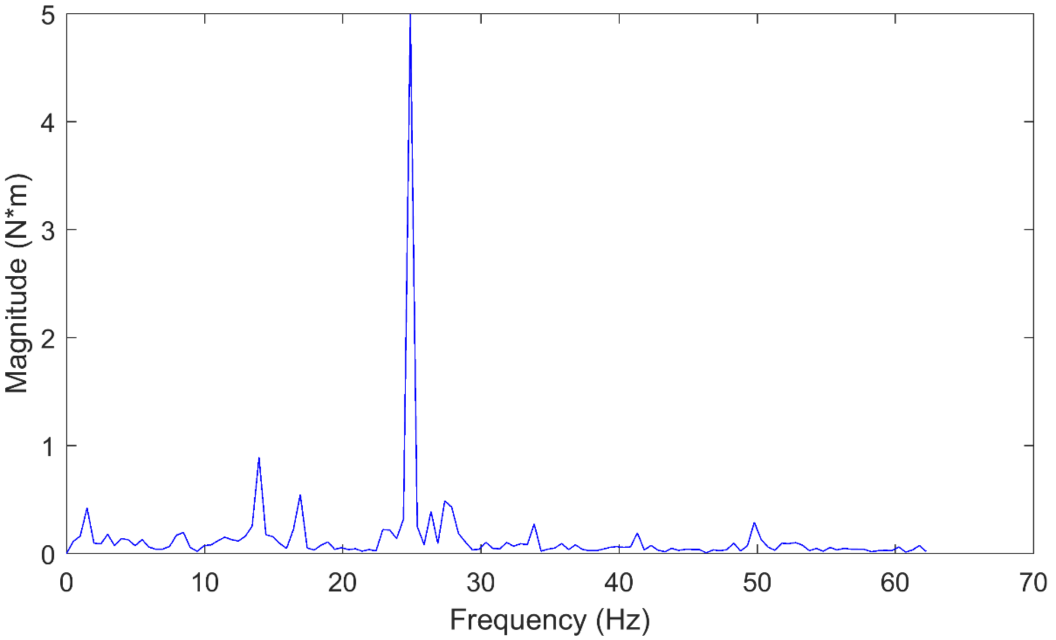


Figure 9.8. FFT of torque data shown in Figure 9.7, with the mean removed. The average engine speed over this time interval is 2991RPM, corresponding to a firing frequency of 24.9Hz. The brake rotor shaft speed over this time interval is 13.9Hz

9.4 Modification of Analytical Model Parameters

To match the actual operation and validate the model, several parameters and boundary conditions in the analytical model must be modified. With the quantities measured on the dynamometer, it is impossible to measure speed loss. Sheave displacement would need to be measured to determine the geometric reduction ratio. The amount of vibration on the dynamometer would yield inaccurate results for this. For comparison to the analytical model, it is assumed that the speed ratio is equal to the geometric reduction ratio. This is needed to map the corresponding experimental to analytical results. For all analytical results presented in this thesis, it is assumed that the engine operates at 3400RPM with a torque of 18N*m. As the engine speed fluctuates through the CVT operating range, this torque changes. In addition, the slack side tractive tension is not known. From measured geometry and mass properties, the axial force in the CVTech primary may be determined as a function of speed and sheave displacement. This may be used as a boundary condition instead of tension.

9.4.1 CVTech Primary Axial Forces

To determine the forces due to the flyweight geometry, a CAD model is created for the primary pulley using measured dimensions in SolidWorks. A picture of the CAD model is shown in Figure 9.9. The wedge-shaped flyweights in the primary slide along linear ramps on the movable sheave, forcing the orientation of the mass to remain constant through the operating range. Forces due to friction are neglected for this analysis. For any sheave displacement, the axial force due to the centripetal acceleration of the three flyweights is calculated as a function of the angular velocity using the following equation:

$$F_A = 3\, m\, r \frac{\Delta D_R}{\Delta D_A}\, \omega^2 \tag{9.2}$$

where m is the mass of each flyweight (228g), r is the radius of the flyweight, and $\frac{\Delta D_R}{\Delta D_A}$ is the ratio of flyweight displacement in the radial direction to axial sheave displacement, or the mechanical advantage of the flyweights. A plot of the flyweight center of mass radius vs pulley nominal radius is shown in Figure 9.10. The nominal radius is calculated from the sheave displacement and the sheave angle. The datasheet for the spring in the primary lists the maximum force at full shift as 950N. Using the change in displacement, the spring constant is 18200N/m. The radius of the belt at full shift is 93mm, yielding the following equation for axial force as a function of primary nominal radius in meters and angular velocity:

$$F_A = 3\,m\,r\frac{\Delta D_R}{\Delta D_A}\,\omega^2 - \big(950 - 18200 * 2\tan(\beta)\,(0.093 - R_P)\big) \tag{9.3}$$

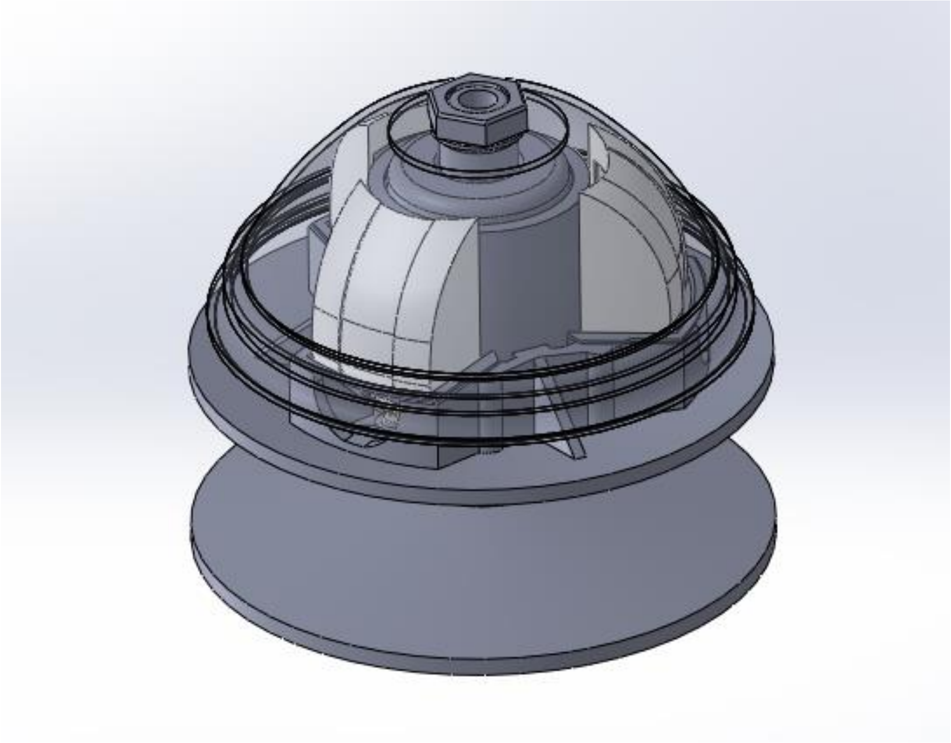


Figure 9.9. CAD model of the CVTech primary pulley. The cap is transparent to illustrate the flyweight geometry.

The nominal primary radius is calculated from the experimental data using the speed ratio, assumed to be equal to the geometric gear ratio, and the measured belt dimensions using Equation 6.12. The MATLAB function used for this is given in the Appendix. Using the measured angular velocity values, the primary pulley axial forces are overlaid on the plot of axial force vs slack span tractive tension, and this is shown in Figure 9.11. This gives insight into the effects on tension of changing the boundary condition to the measured axial forces. The axial forces in the pulley seem to indicate slack side tractive tensions between 100 and 200N.

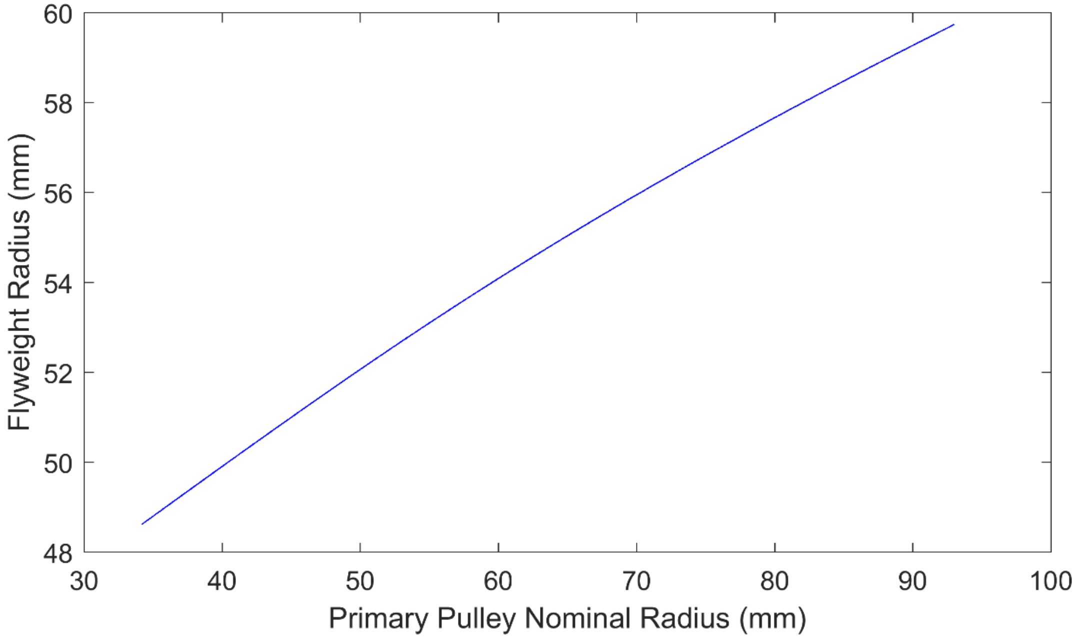


Figure 9.10. Plot of the flyweight radius vs primary pulley nominal radius. This relationship is determined through the CAD model, and the flyweight radius is measured to the center of mass of the flyweights.

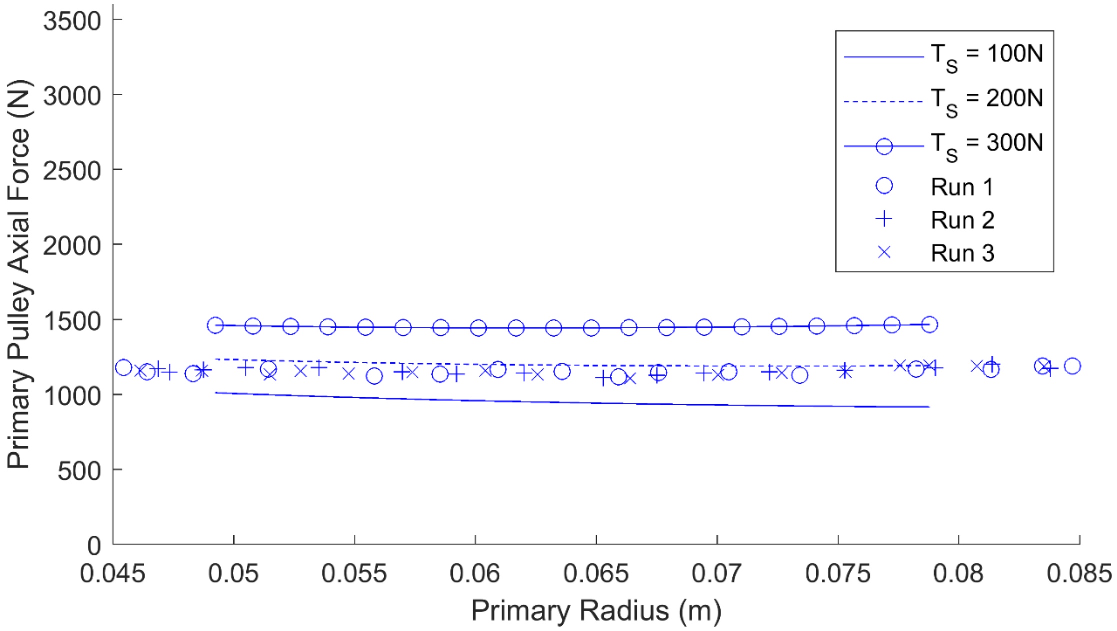


Figure 9.11. Primary axial forces as calculated from experimental results. The results are overlaid on the analytical results for axial force at constant slack span tractive tensions.

9.4.2 Engine Torque

The torque experienced by the primary pulley depends on the engine output. This varies with speed, and the torque is determined through the engine data shown in Figure 9.2. The GRABIT function is used again to create a data vector from the plot. The data vector is interpolated to the primary pulley speed data to determine the torque associated with each data point. To determine the corresponding boundary condition to use in the model, speed ratios are used. The relationship between torque and speed ratio, or primary pulley nominal radius, is interpolated for the radii corresponding to the model results. This is repeated separately for each of the three data runs, and the torque is averaged to find the boundary condition to use at a given primary pulley nominal radius. This same method is used to determine the corresponding axial forces for the model.

9.5 Validation of Analytical Model

Plots of the primary pulley axial force and torque boundary conditions that are used to represent the experimental operating points are shown in Figures 9.12 and 9.13, respectively.

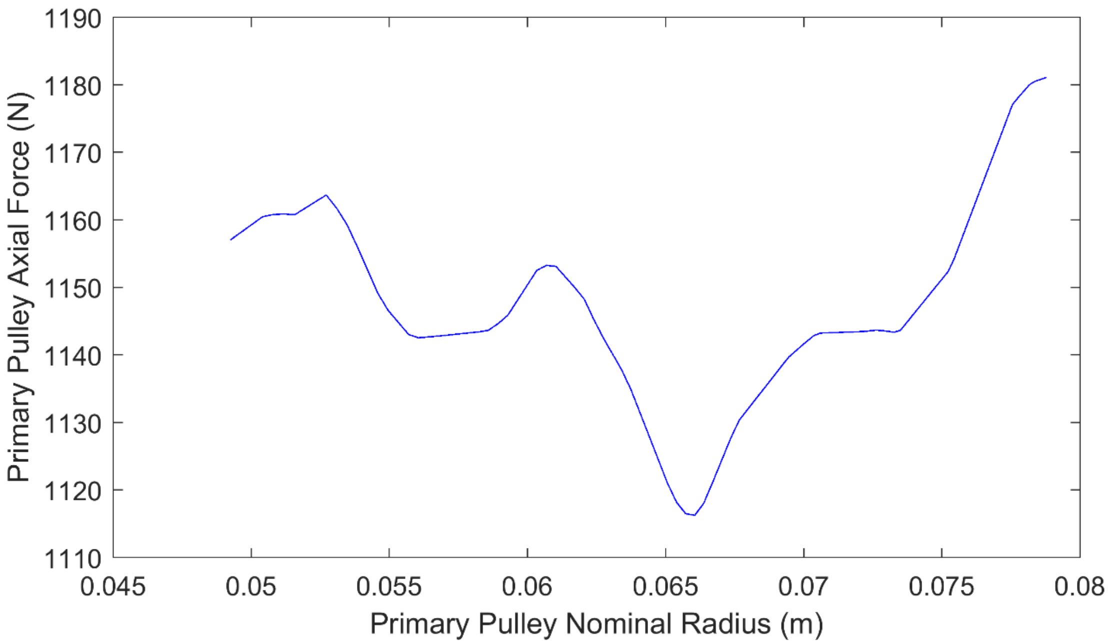


Figure 9.12. Experimental axial force data used as boundary conditions in the analytical model.

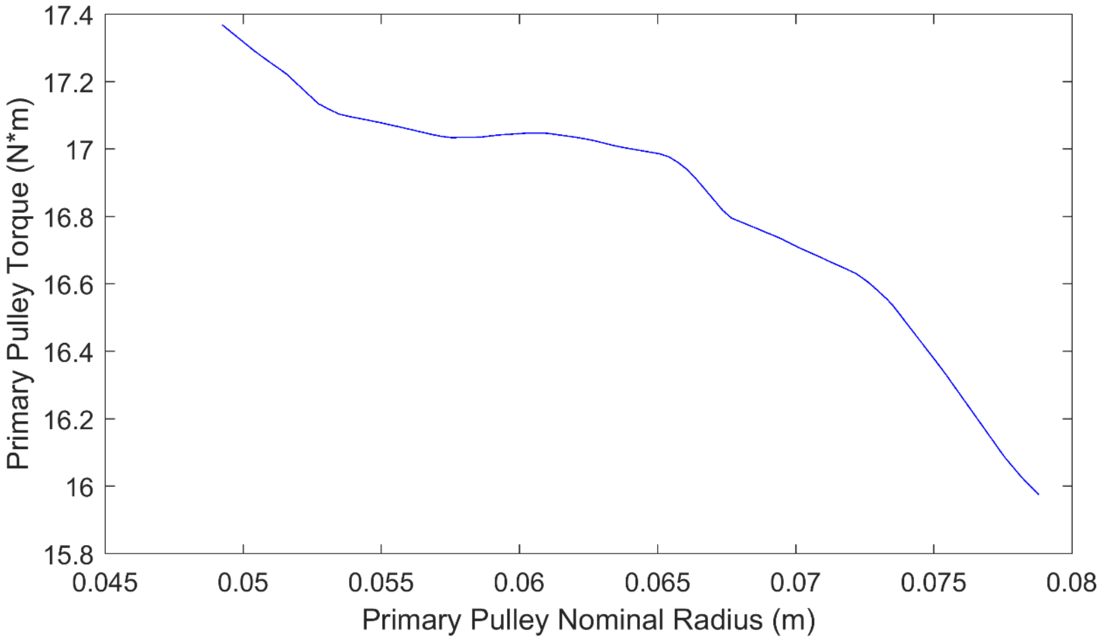


Figure 9.13. Experimental torque data used as boundary conditions in the analytical model.

Continuation methods are again used to adjust the individual solutions representing the operating range to the experimentally determined boundary conditions. The overall efficiency, calculated using the experimentally determined boundary conditions, is shown in Figure 9.14. The data from for all three runs are also shown for comparison. Below a 1:1 ratio, the model agrees well with the data. Both the model and the data indicate that the highest efficiency occurs below a 1:1 ratio, and the efficiency decreases as the system approaches high range. Some of these deviations may be due to inaccuracies in the engine dynamometer data that is used. Since data for the specific engine used is not available, it is possible that the engine used to obtain the dynamometer data may have been older than the engine used here, causing it to output less power due to normal wear and tear.

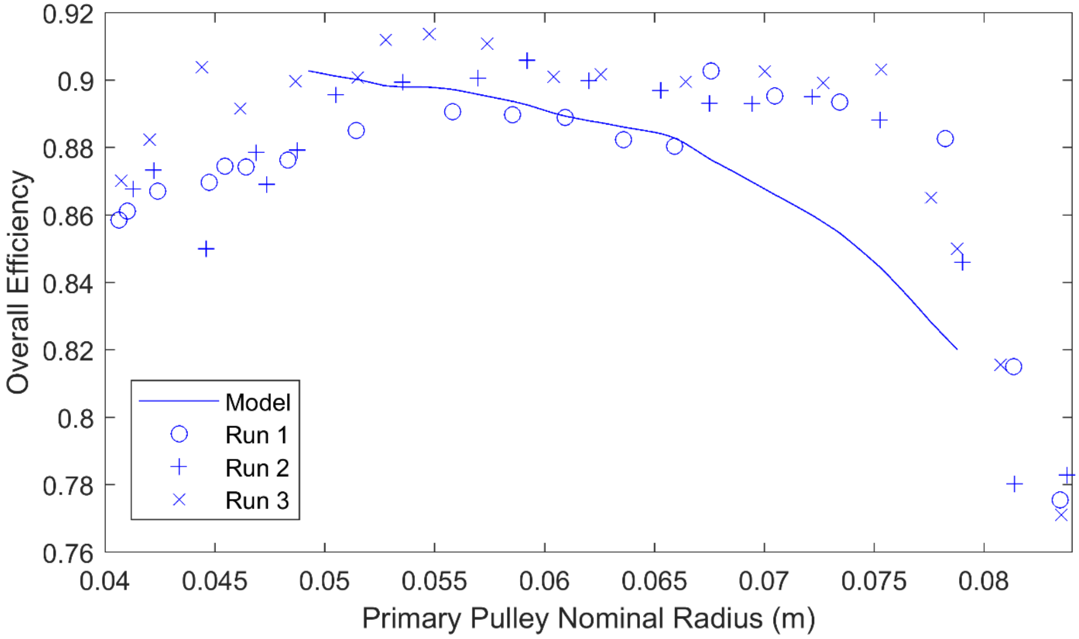


Figure 9.14. Experimental and analytical results for overall efficiency. The model agrees with the data below a 1:1 ratio, (0.068m radius), and both show a decline in efficiency in high range, although the experimental results show a sharper decline in efficiency, at a higher radius than predicted by the model.

10 Conclusion

This thesis improved an existing V-belt string model to calculate pulley axial forces and efficiency, in order to improve the design process for the custom CVT used with the Virginia Tech Baja team. Due to the differences in seating pattern caused by entrance and exit belt tensions, it was shown that the axial force in the primary pulley is substantially higher than that of the secondary at steady-state. Using a viscous-Coulomb friction model allowed the presence of adhesion regions on the pulleys. In the primary pulley, adhesion regions were present, and the length of the region increased as the system approached low range. Using the defined threshold velocity for slip, an adhesion region was never experienced in the secondary pulley, although the relative velocity did decrease as the wrap angle increased. This slippage did not result in excessive losses in speed efficiency due to the low relative magnitude of the sliding velocity.

Adding bending stiffness also improved the accuracy of the solution over existing string models. The bending stiffness had the effect of smoothing the belt path, especially in the entrance and exit regions, and this had the effect of improving solver convergence and extending the range of solutions that are possible. Parameter studies showed the effects of certain parameters on efficiency, and indicate that the overall efficiency can be improved by increasing the sheave angle, decreasing the belt bending stiffness, and decreasing the belt tension in the slack span by decreasing axial force. Overall efficiency did not show a strong dependence on the friction coefficient between the belt and the sheaves.

The construction of the CVT dynamometer allowed for validation of the analytical results for efficiency. The experimental and analytical results agreed well within the range of solutions obtained by the analytical model. Both results showed a peak in efficiency slightly below a 1:1 reduction ratio. This dynamometer will also aid in the tuning process for the custom CVT's designed by the Baja team. Improving the understanding of belt dynamics and providing a method to test and tune the custom designs will ultimately lead to an improved driveline and better results at competition for the Baja team.

11 Future Work

Numerical solutions for the boundary value problem presented in this thesis are obtained using bvp4c, one of the built-in boundary value problem solves in MATLAB. This code for this function is proprietary and is not easily modified. Instability in the governing equations and the nature of the solver forces continuation steps to be small for a solution to be obtained. It may be worth investigating possible simplifications to the equations, or using other available boundary value problem solvers to reduce computation time and improve stability.

All results presented for this thesis are at steady-state. The nature of the boundary value problem presented here does not allow for its extension to a transient case. It may be used to predict a steady state operating point given pulley forces, but it cannot estimate the time taken to reach that point. Several assumptions, such as a constant mass flow rate, G, and a velocity in line with the belt trajectory, would no longer hold and add significant complexity to the model. Aaen presents some qualitative assertions on the effects of certain design decisions on the transient response of a CVT [4]. To thoroughly evaluate a design, more quantitative results are needed.

Other effects worth consideration are due to the inconsistencies in the engine output. Data collected in this thesis indicates that the engine torque varies significantly with crank angle. Due to its steady-state nature, the model presented here cannot predict belt vibration, although this effect is often significant in practice. This vibration may be due to excitation from the oscillating engine output. In addition, losses may be introduced due to the viscous properties of the rubber belt. If these are included, the belt spans may be modeled as a spring and dashpot in parallel. This damping effect, coupled with shock from the engine firing, may result in additional power losses. The magnitude of these shock loads translate to loads experienced by the gearbox attached to the secondary pulley, and these must be taken into consideration for gear design to prevent fatigue failure.

References

[1] SAE International, "2017 Collegiate Design Series Baja SAE Rules," SA International, 2017.

[2] CVTech-IBC, *Mini-Baja Sponsorship Program,* Quebec, 2012.

[3] M. Cammalleri, "A New Approach to the Design of a Speed-Torque-Controlled Rubber V-Belt Variator," *IMechE,* no. 219, pp. 1413-1427, 2005.

[4] O. Aaen, Olav Aaen's Clutch Tuning Handbook, Aaen Performance, 2015.

[5] J.-S. Plante and G. Julio, "An Experimentally-Validated Model of Rubber-Belt CVT Mechanics," *Mechanism and Machine Theory,* no. 46, pp. 1037-1053, 2011.

[6] L. Kong and R. G. Parker, "Mechanics and Sliding Friction in Belt Drives with Pulley Grooves," *Journal of Mechanical Design,* vol. 128, pp. 494-502, 2006.

[7] G. Gerbert, Traction Belt Mechanics, Sweden: Chalmers University of Technology, 1999.

[8] D. Yu, T. H. C. Childs and K. W. Dalgarno, "Experimental and Finite Element Studies of the Running of V-Ribbed Belts in Pulley Grooves," *Proc Instn Mech Engrs,* vol. 212, pp. 343-354, 1998.

[9] O. Reynolds, "On the Efficiency of Belt Straps as Communicators of Work," *The Engineer,* vol. 38, no. 27, pp. 142-145, 1874.

[10] J. L. Meriam and L. G. Kraige, Statics, John Wiley & Sons, Inc., 2012.

[11] L. Kong and R. G. Parker, "Steady Mechanics of Belt-Pulley Systems," *Journal of Applied Mechanics,* vol. 72, pp. 25-34, 2005.

[12] G. Gerbert and F. Sorge, "Full-Sliding Adhesive-Like Contact of V-Belts," *Journal of Mechanical Deisgn,* vol. 124, pp. 706-712, 2002.

[13] D. Karnopp, "Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems," *Journal of Dynamic Systems, Measurement and Control,* vol. 107, pp. 100-103, 1985.

[14] D. Roylance, "Engineering Viscoelasticity," Massachussetts Institute of Technology, Cambridge, 2001.

[15] The Gates Rubber Company, "Temperature Effects on Rubber Belt Operations," The Gates Rubber Company, Denver.

[16] DuPont, "Kevlar Aramid Fiber Technical Guide," DuPont, Richmond, 2017.

[17] L. F. Shampine, I. Gladwell and S. Thompson, Solving ODE's with MATLAB, Cambridge: University of Cambridge, 2003.

[18] MathWorks, "Solve Stiff ODE's," 2018. [Online]. Available: https://www.mathworks.com/help/matlab/math/solve-stiff-odes.html.

[19] U. Ascher and R. D. Russell, "Reformulation of Boundary Value Problems int "Standard" Form," *SIAM Review,* vol. 23, no. 2, pp. 238-254, 1981.

[20] MathWorks, "deval," 2018. [Online]. Available: https://www.mathworks.com/help/matlab/ref/deval.html.

[21] MathWorks, "bvpset," 2018. [Online]. Available: https://www.mathworks.com/help/matlab/ref/bvpset.html.

[22] Melexis, "US5881 Unipolar Hall Switch, Low Sensitivity," 2012.

[23] National Instruments, "Measuring Strain with Strain Gages," 25 May 2016. [Online]. Available: http://www.ni.com/white-paper/3642/en/.

[24] Omega, "Positioning Strain Gages to Monitor Bending, Axial, and Torsional Loads," [Online]. Available: https://www.omega.com/faq/pressure/pdf/positioning.pdf.

[25] Micro Measurements, "187UV General Purpose Strain Gages - Shear/Torque Pattern," VPG, 2016.

[26] Analog Devices, "AD8227 Wide Supply Range, Rail-to-Rail Output Instrumentation Amplifier," 2009.

[27] M. Gerstenhaber and S. Lee, "Strain Gage Measurement using an AC Excitation," Analog Devices, 2004.

[28] Stanford Research Systems, "About Lock-In Amplifiers".

[29] Analog Devices, "AD630 Balanced Modulator/Demodulator," 2016.

[30] Analog Devices, "Analog Filter Wizard," 2018. [Online]. Available: http://www.analog.com/designtools/en/filterwizard/.

[31] Analog Devices, "AD7171, 16-Bit Low Power, Sigma-Delta ADC," 2016.

[32] J. Blum, Exploring Arduino, Indianapolis: John Wiley & Sons, Inc., 2013.

[33] D. Molloy, Exploring Raspberry Pi, Indianapolis: John Wiley & Sons, Inc., 2016.

[34] "forums.bajasae.net," Baja SAE, 2012. [Online]. Available: http://forums.bajasae.net/forum/briggs-dynos-if-your-interested_topic946.html.

[35] MathWorks, "GRABIT," 2016. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/7173-grabit.

Appendix A: Code for Equivalent Cross-Section in Bending

```
%% Bending_Equivalent_Cross_Section.m
% Script to calculate the equivalent cross section of the belt when in
% bending.  The second area moment is obtained through FEA and given as
% an input.  The height of the cross section is calculated numerically
% to match this value.  All dimensions are in meters.
clear
clc
close all
format long

I_bar = 2.019*10^-9;     %second area moment in m^4, estimated from FEA

syms t h A_1 A_2 C_1 C_2 C_r I_bar1 I_bar2 I_1C I_2C

%% Define Equations
% top of trapezoid length
f_1 = (.03051-2*h*tand(12))== t;
% area of square section
f_2 = (.03051*.00285) == A_1;
% area of trapezoidal section
f_3 = ((h/2)*(.03051+t)) == A_2;
% centroid of square section
f_4 = (.00285/2) == C_1;
% centroid of trapezoidal section
f_5 = (.00285+(h/3*(2*t+.03051)/(t+.03051))) == C_2;
% centroid of overall cross-section
f_6 = ((C_1*A_1+C_2*A_2)/(A_1+A_2)) == C_r;
% second area moment of square section about centroid
f_7 = (.03051*.00285^3/12) == I_bar1;
% second area moment of trapezoidal section about centroid
f_8 = (h^3*(t^2+4*t*.03051+.03051^2)/(36*(t+.03051))) == I_bar2;
% square second area moment about overall centroid
f_9 = (I_bar1+A_1*(C_1-C_r)^2) == I_1C;
% trapezoid second area moment about overall centroid
f_10 = (I_bar2+A_2*(C_2-C_r)^2) == I_2C;
% overall second area moment
f_11 = (I_1C+I_2C) == I_bar;

%% Solve Equations
% array of equations to solve
f = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_10, f_11];
% limit all variables to positive real numbers
XO = [0 10^10;0 10^10;0 10^10;0 10^10;0 10^10;0 10^10;0 10^10;0 10^10;...
    0 10^10;0 10^10;0 10^10];

sol = vpasolve(f,[t h A_1 A_2 C_1 C_2 C_r I_bar1 I_bar2 I_1C I_2C],XO);

%% Convert Variables to Double
% thickness at top of belt
t = double(sol.t);
```

```
% height of trapezoidal area in cross section
h = double(sol.h)
% area of square section
A_1 = double(sol.A_1);
% area of trapezoidal section
A_2 = double(sol.A_2);
% centroid height of square section
C_1 = double(sol.C_1);
% centroid height of trapezoidal section
C_2 = double(sol.C_2);
% overall centroid height
C_r = double(sol.C_r)
% second area moment of square section, about centroid
I_bar1 = double(sol.I_bar1);
% second area moment of trapezoidal section, about centroid
I_bar2 = double(sol.I_bar2);
% second area moment of square section, about overall centroid
I_1C = double(sol.I_1C);
% second area moment of trapezoidal section, about overall centroid
I_2C = double(sol.I_2C);



%% Generate Plot
points_x = 1000*[0 0 .03051 .03051 0 (.03051-t)/2 .03051-(.03051-t)/2
.03051];
points_y = 1000*[.00285 0 0 .00285 .00285 h+.00285 h+.00285 .00285];

shadow_x = 1000*[(.03051-t)/2 .01136*tand(12) (.03051-.00775)/2 ...
    (.03051-.00635)/2 .03051-(.03051-.00635)/2 .03051-(.03051-.00775)/2 ...
    .03051-.01136*tand(12) .03051-(.03051-t)/2 .03051-(.01421-.00285-
.00655)...
    *tand(12) (.01421-.00285-.00655)*tand(12) (.03051-t)/2];
shadow_y = 1000*[h+.00285 .01421 .01421 .01221 .01221 .01421 ...
    .01421 h+.00285 .01421-.00655 .01421-.00655 h+.00285];

plot(points_x,points_y)
hold on
plot(shadow_x,shadow_y,':')
hold on
axis equal

hold on
plot(1000*[0 .03051],1000*[C_r C_r],'--')
xlabel('mm')
ylabel('mm')
legend('Equivalent Cross Section','Belt Dimensions','Neutral
Axis','location','SE')
```

Appendix B: Code to Calculate Pulley Nominal Radii

```
function [R_P] = nominal_primary_radius(R_S,L,L_o)
% Function to calculate the primary pulley nominal radius.  Inputs are the
% secondary pulley nominal radius, R_S, the belt center-to-center length,
% L, and the belt pitch length, L_o.
syms y x
%% Solve Equation
% x is the symbolic variable for R_P, adjust to set y equal to 0.
y = 2*(L^2-(R_S-x)^2)^.5 + pi*(x+R_S)+2*asin((R_S-x)/L)*(R_S-x)-L_o;
R_P = vpasolve(y,x);
% Convert solution to double.
R_P = double(R_P);
end
```

```
function [R_P] = radii_given_ratio(ratio,L,L_o)
% Function to calculate the primary pulley nominal radius.  Inputs are the
% ratio, R_S/R_P, the belt center-to-center length, L, and the belt pitch
% length, L_o.
syms y x
%% Solve Equation
% x is the symbolic variable for R_P, adjust to set y equal to 0.
% ratio = R_S/x
% ratio = P_RPM/S_RPM
% R_S = ratio*x
y = 2*(L^2-(ratio*x-x)^2)^.5 + pi*(x+ratio*x)+2*asin((ratio*x-x)/L)*(ratio*x-
x)-L_o;
R_P = vpasolve(y,x);
% Convert solution to double.
R_P = double(R_P);
end
```

Appendix C: Code for Governing Equations (ODE Function)

```
function rhs = rhs_bend_lin_vec(phi,z)
% Function containing the ODE system for the full belt model with
% linearized bending stiffness.  The equations are vectorized to iprove
% performance.

% Subscripts _S indicate values for the secondary, and _P indicate values
% for the primary.
% Subscripts _ts indicate values for the tight span, and _ss indicate
% values for the slack span.
% Prefixes d_ indicate a non-dimensional first derivative.  This is with
% respect to wrap angle in the contact regions, and with respect to arc
% length in the spans.  Prefixes d2_ indicate a second derivative.

%% Import global variables
global EA EI k m_o R_S R_P w_P mu_nom beta slip_threshold

%% Rename solution variables for convenience
% Secondary contact region variables
T_S = z(1,:);
theta_S = z(2,:);
d_theta_S = z(3,:);
d2_theta_S = z(4,:);
x_S = z(5,:);
d_x_S = z(6,:);
w_S = z(7,:);
tau_S = z(8,:);
psi_S = z(9,:);
C_S = z(19,:);
% Primary contact variables
T_P = z(10,:);
theta_P = z(11,:);
d_theta_P = z(12,:);
d2_theta_P = z(13,:);
x_P = z(14,:);
d_x_P = z(15,:);
tau_P = z(17,:);
psi_P = z(18,:);
C_P = z(20,:);
% Mass flow rate
G = z(16,:);
% Tight span variables
T_ts = z(21,:);
theta_ts = z(22,:);
d_theta_ts = z(23,:);
d2_theta_ts = z(24,:);
x_ts = z(25,:);
y_ts = z(26,:);
L_ts = z(27,:);
% Slack span variables
T_ss = z(28,:);
theta_ss = z(29,:);
```

```
d_theta_ss = z(30,:);
d2_theta_ss = z(31,:);
x_ss = z(32,:);
y_ss = z(33,:);
L_ss = z(34,:);
% Axial forces
A_S = z(35,:);
A_P = z(36,:);

%% Evaluate trig functions to reduce computation time
% Secondary angle of inclination
cos_theta_S = cos(theta_S);
sin_theta_S = sin(theta_S);
tan_theta_S = tan(theta_S);
% Primary Anle of inclination
cos_theta_P = cos(theta_P);
sin_theta_P = sin(theta_P);
tan_theta_P = tan(theta_P);

%% Algebraic relations for sheave forces (secondary)
% Belt tension
F_S = (T_S+G.^2./m_o)./(1-G.^2./(EA.*m_o));
% Belt velocity
V_S = (T_S+EA).*G./(EA.*m_o-G.^2);
% Magnitude of the relative velocity between the belt and sheave
mag_V_s_S = sqrt((V_S.*sin_theta_S).^2+(V_S.*cos_theta_S-(R_S-x_S).*w_S).^2);
% Slope of the friction coefficient in the "adhesive-like" region
slope_S = mu_nom/slip_threshold;
% Calculate the modified friction coefficient for all mesh points
mu_S = zeros(1,numel(mag_V_s_S));
for n = 1:numel(mag_V_s_S)
    if mag_V_s_S(n) > slip_threshold
        mu_S(1,n) = mu_nom;
    else
        mu_S(1,n) = slope_S*mag_V_s_S(n);
    end
end
% Sliding angle
Numerator_1 = cos_theta_S-(m_o.*w_S./G).*(R_S-x_S)./(1+F_S./EA);
Denominator_1 = sin_theta_S;
mag_1 = sqrt(Numerator_1.^2+Denominator_1.^2);
cos_gamma_S = Denominator_1./mag_1;
sin_gamma_S = Numerator_1./mag_1;
% Apparent sheave angle, calculated from sliding angle
tan_beta_s_S = tan(beta).*cos_gamma_S;
cos_beta_s_S = 1./sqrt(1+(tan_beta_s_S).^2);
sin_beta_s_S = tan_beta_s_S.*cos_beta_s_S;
% Sheave normal pressure
p_S = k.*x_S./(2.*(cos(beta)+mu_S.*sin_beta_s_S));

%% Algebraic relations for sheave forces (primary)
% Belt tension
F_P = (T_P+G.^2./m_o)./(1-G.^2./(EA.*m_o));
% Belt velocity
V_P = (T_P+EA).*G./(EA.*m_o-G.^2);
% Magnitude of the relative velocity between the belt and sheave
```

```
mag_V_s_P = sqrt((V_P.*sin_theta_P).^2+(V_P.*cos_theta_P-(R_P-x_P).*w_P).^2);
% Slope of the friction coefficient in the "adhesive-like" region
slope_P = mu_nom/slip_threshold;
% Calculate the modified friction coefficient for all mesh points
mu_P = zeros(1,numel(mag_V_s_P));
for n = 1:numel(mag_V_s_P)
    if mag_V_s_P(n) > slip_threshold
        mu_P(1,n) = mu_nom;
    else
        mu_P(1,n) = slope_P*mag_V_s_P(n);
    end
end
% Sliding angle
Numerator_2 = cos_theta_P-(m_o.*w_P./G).*(R_P-x_P)./(1+F_P./EA);
Denominator_2 = sin_theta_P;
mag_2 = sqrt(Numerator_2.^2+Denominator_2.^2);
cos_gamma_P = Denominator_2./mag_2;
sin_gamma_P = Numerator_2./mag_2;
% Apparent sheave angle, calculated from sliding angle
tan_beta_s_P = tan(beta).*cos_gamma_P;
cos_beta_s_P = 1./sqrt(1+(tan_beta_s_P).^2);
sin_beta_s_P = tan_beta_s_P.*cos_beta_s_P;
% Sheave normal pressure
p_P = k.*x_P./(2.*(cos(beta)+mu_P.*sin_beta_s_P));

%% Linearized bending stiffness terms (secondary)
% Second derivatve of radial penetration
d2_x_S = psi_S.*(-R_S.*d_theta_S.*(sec(theta_S)).^2+d_x_S.*tan_theta_S+...
    x_S.*d_theta_S.*(sec(theta_S)).^2);
% Transverse shear force
Q_S = -EI./(R_S.^2.*psi_S.^2).*d2_theta_S+EI./(R_S.^3.*psi_S).*d_x_S;
% Non-dimensional derivative of shear multiplied by 1/psi_S
psi_d_Q_S = EI./(R_S.^3*psi_S.^2).*d2_x_S;
% Coefficient of the non-dimensional third derivative of the angle of
inclination
d3_theta_coeff_S = (-EI.*(R_S+2.*x_S))./(R_S.^3.*psi_S.^3.*T_S);
% Non-dimensional third derivative of the angle of inclination
d3_theta_S = (1-(2.*p_S./T_S).*(sin(beta)-mu_S.*cos_beta_s_S.*...
    (cos_gamma_S-sin_gamma_S.*tan_theta_S)).*(R_S-x_S)-psi_d_Q_S./...
    T_S-d_theta_S./psi_S)./d3_theta_coeff_S;

%% Bending stiffness terms (primary)
% Second derivatve of radial penetration
d2_x_P = psi_P.*(-R_P.*d_theta_P.*(sec(theta_P)).^2+d_x_P.*tan_theta_P+...
    x_P.*d_theta_P.*(sec(theta_P)).^2);
% Transverse shear force
Q_P = -EI./(R_P.^2.*psi_P.^2).*d2_theta_P+EI./(R_P.^3.*psi_P).*d_x_P;
% Non-dimensional derivative of shear multiplied by 1/psi_P
psi_d_Q_P = EI./(R_P.^3*psi_P.^2).*d2_x_P;
% Coefficient of the non-dimensional third derivative of the angle of
inclination
d3_theta_coeff_P = (-EI.*(R_P+2.*x_P))./(R_P.^3.*psi_P.^3.*T_P);
% Non-dimensional third derivative of the angle of inclination
d3_theta_P = (1-(2.*p_P./T_P).*(sin(beta)-mu_P.*cos_beta_s_P.*...
    (cos_gamma_P-sin_gamma_P.*tan_theta_P)).*(R_P-x_P)-psi_d_Q_P./...
    T_P-d_theta_P./psi_P)./d3_theta_coeff_P;
```

```
%% Convert span tractive tensions to tensions
F_ts = (T_ts+G.^2./m_o)./(1-G.^2./(EA.*m_o));
F_ss = (T_ss+G.^2./m_o)./(1-G.^2./(EA.*m_o));

%% ODE system
% Secondary tractive tension
rhs(1,:) = psi_S.*2.*p_S.*(-
sin(beta).*tan_theta_S+mu_S.*cos_beta_s_S.*(tan_theta_S...
    .*cos_gamma_S+sin_gamma_S)).*(R_S-x_S)-psi_S.*Q_S+Q_S.*d_theta_S;
% Secondary angle of inclination
rhs(2,:) = d_theta_S;
% First derivative
rhs(3,:) = d2_theta_S;
% Second derivative
rhs(4,:) = d3_theta_S;
% Secondary radial penetration
rhs(5,:) = d_x_S;
% First derivative
rhs(6,:) = d2_x_S;
% Secondary pulley speed
rhs(7,:) = 0;
% Secondary torque
rhs(8,:) = (2.*mu_S.*p_S.*cos_beta_s_S.*sin_gamma_S.*(R_S-
x_S).^2).*psi_S./cos_theta_S;
% Secondary wrap angle
rhs(9,:) = 0;

% Primary tractive tension
rhs(10,:) = psi_P.*2.*p_P.*(-
sin(beta).*tan_theta_P+mu_P.*cos_beta_s_P.*(tan_theta_P...
    .*cos_gamma_P+sin_gamma_P)).*(R_P-x_P)-psi_P.*Q_P+Q_P.*d_theta_P;
% Primary angle of inclination
rhs(11,:) = d_theta_P;
% First derivative
rhs(12,:) = d2_theta_P;
% Second derivative
rhs(13,:) = d3_theta_P;
% Primary radial penetration
rhs(14,:) = d_x_P;
% First derivative
rhs(15,:) = d2_x_P;
% G, mass flow rate
rhs(16,:) = 0;
% Primary torque
rhs(17,:) = (2.*mu_P.*p_P.*cos_beta_s_P.*sin_gamma_P.*(R_P-
x_P).^2).*psi_P./cos_theta_P;

% Primary wrap angle
rhs(18,:) = 0;
% Secondary orientation angle, C_S
rhs(19,:) = 0;
% Primary orientation angle, C_P
rhs(20,:) = 0;

% Tight span tractive tension
```

```
rhs(21,:) = -L_ts.^2.*EI.*d_theta_ts.*d2_theta_ts;
% Tight span angle of inclination
rhs(22,:) = d_theta_ts;
% First derivative
rhs(23,:) = d2_theta_ts;
% Second derivative
rhs(24,:) = T_ts./(L_ts.^2.*EI).*d_theta_ts;
% Tight span x coordinate
rhs(25,:) = cos(theta_ts);
% Tight span y coordinate
rhs(26,:) = sin(theta_ts);
% Tight span arc length
rhs(27,:) = 0;

% Slack span tractive tension
rhs(28,:) = -L_ss.^2.*EI.*d_theta_ss.*d2_theta_ss;
% Slack span angle of inclination
rhs(29,:) = d_theta_ss;
% First derivative
rhs(30,:) = d2_theta_ss;
% Second derivative
rhs(31,:) = T_ss./(L_ss.^2.*EI).*d_theta_ss;
% Slack span x coordinate
rhs(32,:) = cos(theta_ss);
% Slack span y coordinate
rhs(33,:) = sin(theta_ss);
% Slack span arc length
rhs(34,:) = 0;

% Secondary lateral force
rhs(35,:) = k.*x_S./2.*(psi_S.*(R_S-x_S)./cos_theta_S);
% Primary lateral force
rhs(36,:) = k.*x_P./2.*(psi_P.*(R_P-x_P)./cos_theta_P);
% Secondary unstretched length
rhs(37,:) = psi_S.*(R_S-x_S)./cos_theta_S.*1./(1+F_S./EA);
% Primary unstretched length
rhs(38,:) = psi_P.*(R_P-x_P)./cos_theta_P.*1./(1+F_P./EA);
% Tight span unstretched length
rhs(39,:) = L_ts./(1+F_ts./EA);
% Slack span unstretched length
rhs(40,:) = L_ss./(1+F_ss./EA);

end
```

Appendix D: Code for Boundary Conditions

```
function bc = bc_bend_lin(za,zb)
% Function containing the boundary conditions for the ODE system, defined
% in rhs_bend_lin_vec.m.  Slack span tractive tension is used to define the
% operating point.

% Subscripts _S indicate values for the secondary, and _P indicate values
% for the primary.

%% Import global variables
global R_S R_P L T_slack M_P

%% Calculate entry and exit point locations and angles to the horizon
% Exit angles with respect to positive x-axis
E_St = pi/2-za(19)-zb(2);
E_Ss = -3*pi/2+(3*pi-(za(19)+za(9)))-za(2);
E_Pt = za(20)-pi/2-za(11);
E_Ps = za(20)+za(18)-3*pi/2-zb(11);
% X and y components of secondary exit points
x_St = L-R_S*cos(za(19));
y_St = R_S*sin(za(19));
x_Ss = L-R_S*cos(za(19)+za(9));
y_Ss = R_S*sin(za(19)+za(9));
% X and y components of primary exit points
x_Pt = R_P*cos(za(20));
y_Pt = R_P*sin(za(20));
x_Ps = R_P*cos(za(20)+za(18));
y_Ps = R_P*sin(za(20)+za(18));

%% Initialize boundary conditions vector
bc = zeros(40,1);

%% Secondary boundary conditions
% Secondary radial penetration at entry point
bc(1,1) = za(5);
% Secondary radial penetration at exit point
bc(2,1) = zb(5);
% Secondary torque at entry point
bc(3,1) = za(8);
% Secondary equation for derivative of x, derived in RHS
bc(4,1) = za(6)-za(9)*(-(R_S-za(5))*tan(za(2)));

%% Primary boundary conditions
% Primary torque from engine
bc(5,1) = zb(17)-M_P;
% Primary slack tension
bc(6,1) = zb(10)-T_slack;
% Primary radial penetration at entry point
bc(7,1) = za(14);
% Primary radial penetration at exit point
bc(8,1) = zb(14);
```

```
% Primary torque at entry point
bc(9,1) = za(17);
% Primary equation for derivative of x, derived in RHS
bc(10,1) = za(15)-za(18)*(-(R_P-za(14))*tan(za(11)));

%% Enforce continuity in belt tractive tension across solution regions
% Secondary, slack span
bc(11,1) = za(1)-zb(28);
% Secondary, tight span
bc(12,1) = zb(1)-za(21);
% Primary, tight span
bc(13,1) = za(10)-zb(21);
% Primary, slack span
bc(14,1) = zb(10)-za(28);

%% Enforce continuity in belt angle of inclination
% Secondary, slack span
bc(15,1) = E_Ss-zb(29);
% Secondary, tight span
bc(16,1) = E_St+pi-za(22);
% Primary, tight span
bc(17,1) = E_Pt+pi-zb(22);
% Primary, slack span
bc(18,1) = E_Ps-za(29);

%% Enforce continuity in belt curvature (first derivative of theta)
% Secondary, slack span
bc(19,1) = (cos(za(2))/(R_S-za(5))*(1-1/za(9)*za(3)))-zb(30)/za(34);
% Secondary, tight span
bc(20,1) = (cos(zb(2))/(R_S-zb(5))*(1-1/za(9)*zb(3)))-za(23)/za(27);
% Primary, tight span
bc(21,1) = (cos(za(11))/(R_P-za(14))*(1-1/za(18)*za(12)))-zb(23)/za(27);
% Primary, slack span
bc(22,1) = (cos(zb(11))/(R_P-zb(14))*(1-1/za(18)*zb(12)))-za(30)/za(34);

%% Zero shear at the ends of the contact regions, based on linear
approximation
% Secondary, slack span
bc(23,1) = -za(4)+za(6)/R_S;
% Secondary, tight span
bc(24,1) = -zb(4)+zb(6)/R_S;
% Primary, tight span
bc(25,1) = -za(13)+za(15)/R_P;
% Primary, slack span
bc(26,1) = -zb(13)+za(15)/R_P;

%% Match x and y coordinates of entry and exit points
% X coordinate; secondary, tight span
bc(27,1) = x_St-za(25)*za(27);
% Y coordinate; secondary, tight span
bc(28,1) = y_St-za(26)*za(27);
% X coordinate; secondary, slack span
bc(29,1) = x_Ss-zb(32)*za(34);
% Y coordinate; secondary, slack span
bc(30,1) = y_Ss-zb(33)*za(34);
% X coordinate; primary, tight span
```

```
bc(31,1) = x_Pt-zb(25)*za(27);
% Y coordinate; primary, tight span
bc(32,1) = y_Pt-zb(26)*za(27);
% X coordinate; primary, slack span
bc(33,1) = x_Ps-za(32)*za(34);
% Y coordinate; primary, slack span
bc(34,1) = y_Ps-za(33)*za(34);

%% Zero axial force at entry points
% Secondary
bc(35,1) = za(35);
% Primary
bc(36,1) = za(36);

%% Zero unstretched length at beginning of solution regions
% Secondary
bc(37,1) = za(37);
% Primary
bc(38,1) = za(38);
% Tight span
bc(39,1) = za(39);
% Slack span
bc(40,1) = za(40);

end
```

Appendix E: Code for 1:1 Ratio Continuation Loop

```
%% V_bend_lin_continuation_vectorized.m
% Script to implement a continuation loop for the 1:1 ratio case.
clear
clc
close all
close all hidden

%% Create global variables
% Belt properties
global EA EI k m_o L_o
% System geometry
global R_S R_P L w_P mu_nom beta slip_threshold
% Tension
global T_slack M_P

%% Define system constants
% secondary radius [m]
R_S = 0.068309;
% center-to-center distance [m]
L = 0.214;
% belt unstretched length [m]
L_o = 0.8572;
% calculate primary radius [m]
R_P = nominal_primary_radius(R_S,L,L_o);
% belt longitudinal stiffness [N/(mm/mm)]
EA = 243.5*10^3;
% belt bending stiffness [N/m^2]
EI = 0.2602;
% belt lateral stiffness [N/m^2]
k = 18.71*10^6;
% driver pulley rotational speed [rad/s]
w_P = 3400/60*2*pi;
% belt unstretched mass/unit length [kg/m]
m_o = 0.4375;
% coefficient of friction (kinetic)
mu_nom = 0.4;
% sheave half-angle [rad]
beta = 12*pi/180;
% slack side tractive tension [N]
T_slack = 100;
% driver pulley moment [N*m]
M_P = -18;
% threshold velocity for adhesion [m/s]
slip_threshold = 10^-4;

%% Load saved solution
% Import initial guess from .mat data file.  File is saved in same folder.
% See word document for associated parameters.
load('T_s_100.mat')

%% Resolve system
```

```
options = bvpset('RelTol',1e-3,'Nmax',500,'Vectorized','on','Stats','on');
sol = bvp4c(@rhs_bend_lin_vec,@bc_bend_lin,sol,options);
resolve = 'done'

%% Continuation loop
status = sprintf('progress \n 0.0 %%');
progress = waitbar(0,status);
options = bvpset('RelTol',1e-3,'Nmax',500,'Vectorized','on','Stats','off');

num_steps = 10;

% Loop to increase the slack span tractive tension and save the solution at
% given increments.
for T_slack_end = 125:25:750
    T_slack_increment = (T_slack_end-T_slack)/num_steps;
    for n = 1:num_steps
        T_slack = T_slack+T_slack_increment;
        sol = bvp4c(@rhs_bend_lin_vec,@bc_bend_lin,sol,options);
        status = sprintf('progress \n %.1f %%',n/num_steps*100);
        waitbar(n/num_steps,progress,status)
        if n/1-floor(n/100) < .00001
            display = sprintf('T_s %3.1f  Mesh:  %.0f',T_slack,numel(sol.x));
            disp(display)
        end
    end
    filename = sprintf('T_s_%3.0f.mat',T_slack);
    save(filename,'sol')
end
```

Appendix F: Code to Adjust the Operating Range Solutions

```
%% V_bend_lin_output_sweep.m
% Script to use continuation loops on the individual solutions comprosing
% the operating range.
clear
clc
close all
close all hidden
lastwarn('')

%% Create global variables
% Belt properties
global EA EI k m_o L_o
% System geometry
global R_S R_P L w_P mu_nom beta slip_threshold
% Tension
global T_slack M_P

%% Define system constants
% center-to-center distance [m]
L = .214;
% belt unstretched length [m]
L_o = 0.8572;
% belt longitudinal stiffness [N/(mm/mm)]
EA = 243.5*10^3;
% belt bending stiffness used for output file [N/m^2]
EI_file = 0.2602;
% belt lateral stiffness [N/m^2]
k_file = 18.71*10^6;
% driver pulley rotational speed [rad/s]
w_P = 3400/60*2*pi;
% belt unstretched mass/unit length [kg/m]
m_o = 0.4375;
% coefficient of friction (dynamic)
mu_file = 0.4;
% sheave half-angle in output file [rad]
beta_file = 12*pi/180;
% slack side tractive tension from file [N]
T_slack_file = 100;
% driver pulley moment in N*m
M_P = -18;
% threshold velocity for adhesion [m/s]
slip_threshold = 10^-4;

%% Continuation Loops

load('output_T_s_100.mat')

options = bvpset('RelTol',1e-3,'Nmax',1000,'Vectorized','on','Stats','off');

status_sol = sprintf('current progress \n 0.0 %%');
```

```
progress_sol = waitbar(0,status_sol);

status_overall = sprintf('overall progress \n 0.0 %%');
progress_overall = waitbar(0,status_overall);

num_sols = numel(output.sol);     %number of solutions contained in output
file
num_steps = 5;     %number of steps to change parameters for each solution

T_slack_increment = (25)/num_steps;


T_slack = T_slack_file;
EI = EI_file;
beta = beta_file;
mu_nom = mu_file;
k = k_file;

 for T_slack_actual = 125:25:750
    T_slack_increment = (T_slack_actual-T_slack)/num_steps;
    for n = 1:num_sols
        R_S = output.R_S(n);
        R_P = output.R_P(n);
        EI = EI_actual-0.1;
        T_slack = T_slack_actual-25;

        sol.x = output.sol(n).x;
        sol.y = output.sol(n).y;

        for m = 1:num_steps
            T_slack = T_slack+T_slack_increment;
            sol = bvp4c(@rhs_bend_lin_vec,@bc_bend_lin,sol,options);
            if m/1-floor(m/1) < .0001
                status_sol = sprintf('current progress \n %.1f
%%',(m/num_steps)*100);
                waitbar(m/num_steps,progress_sol,status_sol);
            end
        end

        status_overall = sprintf('overall progress \n %3.1f
%%',n/num_sols*100);
        waitbar(n/num_sols,progress_overall,status_overall);

        output.sol(n) = sol;
        force = [sol.y(35,end) sol.y(36,end)];
        [efficiency] = calculate_efficiency(sol);
        output.force(n,1:2) = force;
        output.efficiency(n,1:3) = efficiency(1:3);
    end

    output.T_slack = T_slack;
    filename = sprintf('output_T_s_%3.0f_EI_%1.3f.mat',T_slack,EI);
    save(filename,'output');
 end
```

Appendix G: Arduino Code

```
/*  Dyno slave arduino code to read two simultaneous RPM's
 *  using Hall Effect sensors and to read the AD7171 16 bit A/D converter.
 *  The Arduino will read values continuously and output values with
timestamps
 *  to the Raspberry Pi when called.  A checksum is used to detect errors in
 *  the I2C transmission.
*/

#include <Wire.h>

//digital pin for hall effect sensor
int hall_P_pin = digitalPinToInterrupt(2);
int hall_S_pin = digitalPinToInterrupt(3);
//time of last RPM measurement
volatile unsigned long time_P_last = 0;
volatile unsigned long time_S_last = 0;
//time of current RPM measurement
volatile unsigned long time_P_current;
volatile unsigned long time_S_current;
//last time triggered
volatile unsigned long last_read_P = 0;
volatile unsigned long last_read_S = 0;
//RPM variables
volatile unsigned int P_RPM = 100;
volatile unsigned int S_RPM = 100;

//ADC variables
int CLOCK = 4;
int DATA = 5;
bool error = 0;
int num_bits;
unsigned int value;

//time stamps
volatile unsigned long stamp_P_RPM = 0;
volatile unsigned long stamp_S_RPM = 0;
volatile unsigned long stamp_strain = 0;

//array to pass via i2c
unsigned int sum;
byte buf[20];

void setup() {
  //setup strain gage pins
  pinMode(CLOCK, OUTPUT);
  pinMode(DATA, INPUT_PULLUP);
  digitalWrite(CLOCK, HIGH);
  delay(1);

  //initialize interrupt pins
  attachInterrupt(hall_P_pin, magnet_detect_P, FALLING);
  attachInterrupt(hall_S_pin, magnet_detect_S, FALLING);
  //begin serial communication
  Wire.begin(3);
  Wire.onRequest(send_data);
}

// Functions to calculate rpm
void magnet_detect_P(){
  //get current time
  time_P_current = micros();
  //ignore interupt if less than threshold time (debouncing)
  if (micros() > last_read_P+10000){
    last_read_P = micros();
    P_RPM = 1000000*60/(time_P_current-time_P_last);
    //write values to buffer for transmission
    buf[0] = P_RPM >> 8;
    buf[1] = P_RPM;
    buf[2] = time_P_current >> 24;
    buf[3] = time_P_current >> 16;
    buf[4] = time_P_current >> 8;
    buf[5] = time_P_current;
    time_P_last = time_P_current;
  }
}
void magnet_detect_S(){
  //get current time
  time_S_current = micros();
  //ignore interupt if less than threshold time (debouncing)
  if (micros() > last_read_S+6000){
    last_read_S = micros();
    S_RPM = 1000000*60/(time_S_current-time_S_last);
    //write values to buffer for transmission
    buf[6] = S_RPM >> 8;
    buf[7] = S_RPM;
    buf[8] = time_S_current >> 24;
    buf[9] = time_S_current >> 16;
    buf[10] = time_S_current >> 8;
    buf[11] = time_S_current;
    time_S_last = time_S_current;
  }
}

// Function to send current RPM values to Master Arduino
void send_data() {
  //compute checksum
  sum =
buf[0]+buf[1]+buf[2]+buf[3]+buf[4]+buf[5]+buf[6]+buf[7]+buf[8]+buf[9]+buf[10]
+buf[11]+buf[12]+buf[13]+buf[14]+buf[15]+buf[16]+buf[17];
  buf[18] = sum >> 8;
  buf[19] = sum;
  Wire.write(buf, 20);
}

// Main loop to read strain gages
void loop() {
  //wait for AD7171 to indicate a conversion is ready
  while (digitalRead(DATA)){
  }
  stamp_strain = micros();
  delayMicroseconds(100);

  //send clock pulses and read signal
  value = 0;
  num_bits = 15;
  while (num_bits >= 0){
    digitalWrite(CLOCK, LOW);
    digitalWrite(CLOCK, HIGH);
    bitWrite(value, num_bits, digitalRead(DATA));
    num_bits--;
  }
  //read appended bits to check for errors
  num_bits = 0;
  while (num_bits <= 7){
    digitalWrite(CLOCK, LOW);
    digitalWrite(CLOCK, HIGH);
    if (num_bits == 2){
      error = digitalRead(DATA);
    }
    num_bits++;
  }
  //write values to buffer for transmission if no error has occurred
  if (error == 0){
    buf[12] = value >> 8;
    buf[13] = value;
    buf[14] = stamp_strain >> 24;
    buf[15] = stamp_strain >> 16;
    buf[16] = stamp_strain >> 8;
    buf[17] = stamp_strain;
  }
}
```

Appendix H: Raspberry Pi Code

```
# dyno_main_funcs.py
# Python code containing common functions to be called by other python
scripts.
import smbus
import RPi.GPIO as GPIO
import time
import sys
from pathlib import Path
import csv
from scipy.interpolate import interp1d

MOS_pin = 33
DPDT_pin = 16
servo_pin = 12
kill_pin = 18
reset_pin = 22

def init_I2C():
    # This function initializes the I2C connection using the built-in SMBus
library.
    # Returns the bus, using the default bus 1 on the RPi.
    # Call as:  bus = init_I2C()

    return smbus.SMBus(bus = 1)

def init_GPIO():
    # This function initializes all GPIO pins used on the RPi.
    # Pin numbers are assigned to corrrespond to the header on the PCB.
    # Returns the servo object to control engine throttle.
    # Call as:  servo = init_GPIO()

    # Set GPIO mode to board layout and turn off errors
    GPIO.setmode(GPIO.BOARD)
    GPIO.setwarnings(False)
    # Setup all pins as inputs or outputs
    GPIO.setup(MOS_pin, GPIO.OUT)
    GPIO.setup(DPDT_pin, GPIO.OUT)
    GPIO.setup(servo_pin, GPIO.OUT)
    GPIO.setup(kill_pin, GPIO.OUT)
    GPIO.setup(reset_pin, GPIO.OUT)
    # Define all output initial values
    GPIO.output(MOS_pin, 0)
    GPIO.output(DPDT_pin, 0)
    GPIO.output(kill_pin, 0)
    GPIO.output(reset_pin, 0)
    # Start servo PWM signal
    servo = GPIO.PWM(servo_pin, 50)
    servo.start(11)
    return servo

def set_throttle(servo, val):
    # This function changes the throttle to val.  val must be between 0 and 
100.
    # Returns the servo object.
    # Call as: servo = set_throttle(servo, val)

    if val >= 0 and val <= 100:
        # Map throttle value to servo limits
        duty = interp1d([0, 100], [11, 4])
        servo.ChangeDutyCycle(float(duty(val)))
    return servo

def reset_arduino():
    # This function is used to reset the arduino, and reset the hardware 
clock to zero.
    # This prevents clock rollover (after 70 min) from occurring during data 
collection.
    # No variables are returned.
    # Call as:  reset_arduino()

    GPIO.output(reset_pin, 1)
    # Delay to ensure reset has occurred
    time.sleep(.25)
    GPIO.output(reset_pin, 0)
    # Delay to allow the arduino to initialize
    time.sleep(1)

def get_cal_data():
    # This function opens the calibration file to obtain the current 
calibration values.
    # Returns the slope and intercept from the last run in the calibration 
file.
    # Call as: [cal_b, cal_m] = get_cal_data()

    f = open('/home/pi/Documents/DYNO/FINAL_CODE/cal_values.csv')
    csv_f = csv.reader(f)
    # Reassign values until last row is reached
    for row in csv_f:
        cal_m = float(row[0])
        cal_b = float(row[1])
    return cal_b, cal_m

def apply_brake():
    # This function sets the brake to apply continuously.
    # Use stop_brake() or release_brake() to stop brake application.
    # No variables are returned.
    # Call as: apply_brake()

    # Turn off actuator and set direction to retract
    GPIO.output(MOS_pin, 0)
    GPIO.output(DPDT_pin, 1)
    # Retract actuator
    GPIO.output(MOS_pin, 1)

def release_brake():
    # This function sets the brake to release continuously.
    # Actuator end stops will prevent over-extension.
    # No variables are returned.
    # Call as: release_brake()

    # Turn off actuator and set direction to extend
    GPIO.output(MOS_pin, 0)
    GPIO.output(DPDT_pin, 0)
    # Extend actuator
    GPIO.output(MOS_pin, 1)

def step_brake(time_move):
    # This function increments the brake for a specified time.
    # Use a positive input to apply the brake, negative to release.
    # No variables are returned.
    # Call as: step_brake(<time to move>)

    # Set direction for actuator
    if time_move >= 0:
        GPIO.output(MOS_pin, 0)
        GPIO.output(DPDT_pin, 1)
    else:
        GPIO.output(MOS_pin, 0)
        GPIO.output(DPDT_pin, 0)
    # Move actuator for specified time
    GPIO.output(MOS_pin, 1)
    time.sleep(abs(time_move))
    GPIO.output(MOS_pin, 0)

def read_temp(bus):
    # This function reads the rotor temperature from the IR temp sensor.
    # Returns the rotor temperature in degrees Celsius and a corresponding
color.
    # The color may be used as a window background color.
    # Call as: [temp, colorhex] = read_temp(bus)

    # I2C address and register for sensor
    ir_add = 0x5a
    ir_reg = 0x07
    # Read the temperature and convert to correct units
    try:
        raw = bus.read_word_data(ir_add, ir_reg)
    except:
        raw = 0
    temp = raw*0.02-273.15
    color = interp1d([20, 200],[0, 255])
    if temp < 20:
        colorhex = "#%02x%02x%02x" % (0, 255, 0)
    elif temp > 200:
        colorhex = "#%02x%02x%02x" % (255, 0, 0)
    else:
        colorhex = "#%02x%02x%02x" % (int(color(temp)), 255-int(color(temp)),
0)
    return temp, colorhex

def E_STOP(servo, window):
    # Function to initiate emergency stop for system.
    # Inputs are the throttle servo and main window objects.
    # No variables are returned.
    # Call as: E_STOP(servo, root)
    global status
    status = 'EMERGENCY STOP'
    # Set throttle servo to idle position
    servo.ChangeDutyCycle(11)
    # Kill engine
    GPIO.output(kill_pin, 1)
    # Release brake
    GPIO.output(MOS_pin, 0)
    GPIO.output(DPDT_pin, 0)
    GPIO.output(MOS_pin, 1)
    # Wait for engine to stop
    time.sleep(15)
    window.quit()
    window.destroy()
    GPIO.cleanup()

def get_values(bus):
    # Function to call data points from the Arduino through I2C.
    # Checks transmission for errors and returns a flag if an error occurred.
    # Returns values and timeastamps for P_RPM, S_RPM, and ADC_RAW.
    # Call as: [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)

    arduino_add = 0x03
    arduino_reg = 0
    num_bytes = 20
    # Request data from the Arduino
    try:
        val = bus.read_i2c_block_data(arduino_add, arduino_reg, num_bytes)
    # Return an error if the Arduino does not acknowledge the request
    except:
        val = [0]*num_bytes
        val[0] = 1
    checksum = val[18]*256+val[19]
    # Compare checksum to check for errors
    if sum(val[0:18]) == checksum:
        error = 0
    else:
        error = 1
    # Concatenate bytes to reconstruct 16 and 32 bit values
    P_RPM = val[0]*256+val[1]
    P_time = val[2]*16777216+val[3]*65536+val[4]*256+val[5]
    S_RPM = val[6]*256+val[7]
    S_time = val[8]*16777216+val[9]*65536+val[10]*256+val[11]
    ADC_raw = val[12]*256+val[13]
    ADC_time = val[14]*16777216+val[15]*65536+val[16]*256+val[17]
    return P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error

def convert_ADC_raw_torque(ADC_log, cal_b, cal_m):
    # Function to convert raw ADC data to torque data using calibrated
values.
    # Returns list of torque values with the same timestamps as the original
data set.
    # Call as: TORQUE = convert_ADC_raw_torque(ADC_raw, cal_b, cal_m)

    TORQUE = [(float(x)-cal_b)/cal_m for x in ADC_log]
    return TORQUE

def write_file(time_data, data, filename):
    # Function to save a data file on the mounted flash drive.
    # No variables are returned.
    # Call as: write_file(time_data, data, filename)

    # Define the file path
    directory = '/media/usb1/'
    file = Path(directory+filename+'.csv')
    num_data = len(data)
    # Write data by row
    open(str(file), 'a')
    with open(str(file), 'a') as dataFile:
        writer = csv.writer(dataFile, quoting = csv.QUOTE_NONE)
        line = 0
        while line < num_data:
            writer.writerow([time_data[line], data[line]])
            line = line+1
        dataFile.close
```

```
# steady_state_step.py
# Python code to run steady-state test case
from dyno_main_funcs import *
from tkinter import *
import time
import smbus
import _thread
from scipy.interpolate import interp1d
import os
import glob

P_RPM = 0
S_RPM = 0
ADC_raw = 0
P_time = 0
S_time = 0
ADC_time = 0
error = 0
data_index = 0

time_read = 3
sample_rate = 150.
log_period = 1/sample_rate

cal_b = 0
cal_m = 0

status = 'INITIALIZING'
wait_status = 1

def run(rpm_min):
    global servo, bus
    # Main function to run the steady-state test case.
    # The minimum desired secondary RPM must be passed as an input.

    def delete_files():
        # Function to delete old data files from the flash drive.
        # All files not in a directory will be removed.

        for filename in glob.glob('/media/usb1/P_*'):
            os.remove(filename)
        for filename in glob.glob('/media/usb1/S_*'):
            os.remove(filename)
        for filename in glob.glob('/media/usb1/ADC_*'):
            os.remove(filename)
        for filename in glob.glob('/media/usb1/TORQUE_*'):
            os.remove(filename)

    def wait_button():
        global wait_status
        # Function to execute when the "GO" button is pressed.
        # This sets a flag to allow execution to continue,

        wait_status = 0

    def stst_window():
        global stst, servo, bus, status, wait_status, P_RPM, S_RPM, ADC_raw, 
data_index
        # Function to initialize the GUI.

        bgcolor = 'white'
        stst = Tk()
        stst.status_label = Label(text = str(status)+'\n', font = (None, 36))
        stst.status_label.pack()
        stst.blank_space1 = Label(text = '          ', background = bgcolor, 
font = (None, 36))
        stst.blank_space1.pack()
        stst.P_RPM_label = Label(text = 'Engine RPM', font = (None, 36))
        stst.P_RPM_label.pack()
        stst.P_RPM_value = Label(text = str(P_RPM), font = (None, 36))
        stst.P_RPM_value.pack()
        stst.S_RPM_label = Label(text = 'Secondary RPM', font = (None, 36))
        stst.S_RPM_label.pack()
        stst.S_RPM_value = Label(text = str(S_RPM), font = (None, 36))
        stst.S_RPM_value.pack()
        stst.setpoint_label = Label(text = 'Data Index', font = (None, 36))
        stst.setpoint_label.pack()
        stst.setpoint_value = Label(text = str(data_index), font = (None, 
36))
        stst.setpoint_value.pack()
        stst.strain_label = Label(text = 'Torque (RAW)', font = (None, 36))
        stst.strain_label.pack()
        stst.strain_value = Label(text = str(ADC_raw), font = (None, 36))
        stst.strain_value.pack()
        stst.temp_label = Label(text = 'Rotor Temp (C)', font = (None, 36))
        stst.temp_label.pack()
        [temp, colorhex] = read_temp(bus)
        stst.temp_value = Label(text = str(temp), font = (None, 36))
        stst.temp_value.pack()

        stst.blank_space2 = Label(text = '          ', background = bgcolor, 
font = (None, 36))
        stst.blank_space2.pack()

        stst.wait_go_button = Button(text = 'GO', font = (None, 36), command 
= wait_button)
        stst.wait_go_button.pack()
        stst.blank_space3 = Label(text = '          ', background = bgcolor, 
font = (None, 36))
        stst.blank_space3.pack()
        stst.abortButton = Button(text = 'EMERGENCY STOP', font = (None, 36), 
command = lambda: E_STOP(servo, stst))
        stst.abortButton.pack()
        stst.attributes('-fullscreen', True)

    def update_display():
        global stst, servo, bus, status, wait_status, P_RPM, S_RPM, ADC_raw, 
data_index
        # Function to update the data labels with "live" values.

        try:
            # Update data values
            stst.status_label.config(text = str(status))
            stst.P_RPM_value.config(text = str(P_RPM))
            stst.S_RPM_value.config(text = str(S_RPM))
            stst.strain_value.config(text = str(ADC_raw))
            stst.setpoint_value.config(text = str(data_index))
            # Read temperature of brake rotor
            [temp, colorhex] = read_temp(bus)
            stst.temp_value.config(text = str(round(temp, 1)))
            # Update background color based on the rotor temperature
            stst.configure(bg = colorhex)
            stst.blank_space1.configure(background = colorhex)
            stst.blank_space2.configure(background = colorhex)
            stst.blank_space3.configure(background = colorhex)
            stst.after(1000, update_display)
        except:
            pass

    def test_thread():
        global stst, servo, bus, status, wait_status, P_RPM, S_RPM, ADC_raw,
data_index
        # The GUI window is an infinite loop and blocks further execution in
the main thread.
        # This thread allows execution of the test code to continue in the
background.
        # No inputs are passed and no variables are returned.

        def write_data(cal_b, cal_m, P_time_log, P_log, S_time_log, S_log,
ADC_time_log, ADC_log):
            # This function writes three separate data files for each brake
increment.
            # Calibration values and data lists are passed as inputs.

            write_file(P_time_log, P_log, 'P_'+str(data_index))
            write_file(S_time_log, S_log, 'S_'+str(data_index))
            write_file(ADC_time_log, ADC_log, 'ADC_'+str(data_index))
            # Convert raw ADC values to torque using calibration data
            TORQUE = convert_ADC_raw_torque(ADC_log, cal_b, cal_m)
            write_file(ADC_time_log, TORQUE, 'TORQUE_'+str(data_index))

        servo = set_throttle(servo, 0)
        # Get calibration data from file
        [cal_b, cal_m] = get_cal_data()
        # Call values from arduino
        [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
        # Retry transmission until successful
        while error == 1:
            [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
        # Move brake to initial position
        step_brake(3)
        # Wait for engine to start
        status = 'WAITING FOR ENGINE'
        while P_RPM < 1300:
            [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
            while error == 1:
                [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
            time.sleep(0.1)
        # Wait for user input to begin test
        wait_status = 1
        status = 'PRESS GO TO RUN'
        while wait_status:
           pass
        wait_status = 1
        # Set throttle to full
        servo = set_throttle(servo, 100)
        status = 'INCREASING THROTTLE'
        # Wait for system to stabilize
        for x in range(0, 10):
            [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
            while error == 1:
                [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
            time.sleep(1)
        # Continue test until secondary RPM decreases below the minimum
desired value
        while S_RPM >= rpm_min or S_RPM == 0:
            # Initialize logging variables
            P_log = []
            P_time_log = []
            S_log = []
            S_time_log = []
            ADC_log = []
            ADC_time_log = []
            # Increment brake
            step_brake(0.05)
            # Lock brake and allow system to settle
            status = 'LOCKED'
            for x in range(0, 4):
                [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
                while error == 1:
                    [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error]
= get_values(bus)
                time.sleep(1)
            # Record data for specified time interval
            status = 'READING'
            time_next = time.time()+time_read
            time_log = time.time()+log_period
            while time.time() <= time_next:
                while time.time() <= time_log:
                    pass
                [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
                # Append values to logging array if transmission was
successful
                if error == 0:
                    P_log.append(P_RPM)
                    P_time_log.append(P_time)
                    S_log.append(S_RPM)
                    S_time_log.append(S_time)
                    ADC_log.append(ADC_raw)
                    ADC_time_log.append(ADC_time)
                    time_log = time_log+log_period
            # Write data arrays to files
            write_data(cal_b, cal_m, P_time_log, P_log, S_time_log, S_log,
ADC_time_log, ADC_log)
            data_index = data_index+1
            while error == 1:
                [P_time, P_RPM, S_time, S_RPM, ADC_time, ADC_raw, error] =
get_values(bus)
        status = 'TEST_COMPLETED'
        E_STOP(servo, stst)

    # Delete files from previous data runs
    delete_files()
    # Initialize GPIO and I2C
    bus = init_I2C()
    servo = init_GPIO()
    reset_arduino()
    _thread.start_new_thread(test_thread, ())
    stst_window()
    update_display()
    stst.mainloop()

if __name__ == '__main__':
    run(1200)
```