Modelling of a motorcycle/mountain bike suspension and digitization of a cam drum motor control

RASMUS NORD
Modelling of a Motorcycle/Mountain Bike Suspension and Digitization of a Cam Drum Motor Control

Rasmus Nord
ranord@kth.se

Company supervisor: Carl Hesse, Öhlins Racing AB
Supervisor: Jenny Jerrelind, KTH Vehicle Dynamics
Examiner: Jenny Jerrelind, KTH Vehicle Dynamics
Abstract
This Master's thesis was done in collaboration with Öhlins Racing AB, a Swedish suspension system-manufacturer. For Öhlins, the quality of their products is highly important and they are therefore devoting many resources for the development and testing of their products. Because testing is such a big part of what Öhlins as a company is doing, it is important to continuously strive to improve the testing methods used within the company.

Two popular methods for improving testing methods are through automation of the machine control and by simulating the test with mathematical models. Both methods have the potential to reduce the time consumed during testing. This project focuses on these two methods and is therefore split in two parts. The first part focuses on digitizing the motor control of a rolling road test bench called the Cam Drum, which is used to do life cycle tests of suspension assemblies, to allow for automated control. In the second part the rolling road test bench has been modelled as a suspension system to simulate tests prior to production. The goal of the digitization is to enable more advanced tests while simplifying usage of the Cam Drum, thereby reducing the time necessary to operate the machine. The goal of the suspension model is to get validation results that point towards the model being good enough to use as a tool when developing new products.

A programmable logic controller was connected to the existing frequency drive that controls motor rotational speed and an HMI screen was used to control the controller. Communication between the controller and frequency drive used the serial protocol Modbus RTU. The hardware with which the new motor control system was built was primarily supplied by Siemens. Controller and HMI programming was carried out in Siemens' software SIMATIC Step 7 using programming languages LAD and FBD. The digital motor control system was live tested with great results and good feedback from the technicians. The only functionality missing is being able to send webserver data over the buildings industrial network due to IT related security reasons. Future work should focus on solving this problem.

A front fork suspension model and a rear swingarm suspension model have been modelled in Matlab Simulink. Both models are designed to simulate motorcycle or mountain bike suspension however the front suspension model has only been validated against mountain bike data and the rear suspension model against motorcycle data.

An alternative tire model was developed to handle problems linked to conventional 1-dimensional tire models. The new model estimates the area of compressed air in the side view plane and scales the force output accordingly. New values for tire spring stiffness and damping coefficient for this system was freely estimated during validation.

Validation was done using camera recorded position signals and position signals recorded with a position sensor. The front suspension model was tested against two different front fork models, but validation finally focused on several test runs done with one of the forks due to insufficient recorded data with the other fork. The result was a correlation between the behaviour of the real and modelled suspension however further tweaking of the tire parameters should give better results. The result should however be sufficient for making estimations.

Validation of the rear suspension was done against a camera recorded position signal but as evidence from the front suspension validation shows this is insufficient. The rear suspension validation still requires more work before being utilized as a development tool.
Sammanfattning

Detta examensarbete är utfört i samarbete med Öhlins Racing AB, som är ett svenskt hjulupphängningsföretag. Kvalité är mycket viktigt för Öhlins, därför lägger de mycket resurser på utveckling och provning av deras produkter. I och med att provning är en så stor del av vad Öhlins som företag gör så är det viktigt att kontinuerligt sträva efter att förbättra de provmetoder som används inom företaget.


En framgaffel och en bakhjulsupphängning har modellerats i Matlab Simulink. Båda modellerna är designade för att simulera en motorcykel eller mountain bike-upphängning men framgaffelmodellen har endast validerats mot data från mountain bikes och baksvingsmodellen har endast validerats mot motorcykeldata.

En alternativ däckmodell har tagits fram för att åtgärda problem kopplade till konventionella endimensionella däckmodeller. Den nya modellen estimerar arean av den komprimerade luften sett från sidan och skalar kraften den producerar enligt arean. Nya värden för däckets fjäderstyrhet och dämparkoefficient för det här systemet har tagits fram fritt under valideringen.

Baksvingsmodellen validerades enbart mot en positionssignal från en video vilket resultatet från framgaffelmodellens validering tyder på är otillräckligt. Mer jobb måste läggas på validering av bakhjulupphängningen innan den kan användas i utvecklingssyfte.
## Index of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example of a mountain bike rear suspension layout [2]</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Example of a motorcycle rear suspension layout [3]</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Conventional 2-DOF mass-spring-damper system</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Adjusted 2-DOF mass-spring-damper system</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Illustration of the Cam Drum setup of a front suspension with relevant geometrical variables</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Illustration of the Cam Drum setup of a rear suspension with relevant geometrical variables</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>Geometrical illustration of the rear suspension swingarm</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Comparative sketch of the rear swingarm in its original position and at a point during the simulation. The line between points a and c represents the swingarm during the simulation and the line between a₀ and c represents the swingarm in its original position</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Isolated sketch of the geometrical layout of the rear coilover</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>Illustration of how using two points to calculate tire forces can be problematic</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>Lateral view of the wheel and road profile</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Cross section of the tire</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>Cam drum fork geometry in steady state (solid lines) and after hitting a bump (dashed lines)</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>Wheel lift logic of the model</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>Comparison between recorded Tracker data from the first data recording on front fork #1 at 12 km/h (tangential speed) and model simulation positions after some validation work</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>Comparison between recorded Tracker data from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation positions</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>Comparison between recorded position sensor data (difference in spring and damper length from steady state) from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation compression and rebound</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>Comparison between recorded Tracker data from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation positions using the updated tire model</td>
<td>17</td>
</tr>
<tr>
<td>19</td>
<td>Comparison between recorded position sensor data (difference in spring and damper length from steady state) from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation compression and rebound using the updated tire model</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>Comparison between recorded spring and damper velocity from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation compression and rebound using the updated tire model. Both signals filtered using a Butterworth filter</td>
<td>18</td>
</tr>
<tr>
<td>21</td>
<td>Comparison between recorded position sensor data (difference in spring and damper length from steady state) from the second data recording on front fork #2 at 45 km/h (tangential speed) and model simulation compression and rebound using the updated tire model</td>
<td>19</td>
</tr>
<tr>
<td>22</td>
<td>Comparison between recorded spring and damper velocity from a data recording on front fork #2 at 45 km/h (tangential speed) and model simulation compression and</td>
<td>20</td>
</tr>
</tbody>
</table>
rebound velocity using the updated tire model. Both signals filtered using a Butterworth filter.

Figure 23. Comparison between recorded Tracker data from the first data recording on the swingarm at 18 km/h (tangential speed) and model simulation positions after some validation work.

Figure 24. Network 1: Configuration of communication port.

Figure 25. Network 2: Modbus Master block.

Figure 26. The root screen of the HMI program.

Figure 27. The PLC test program screen for testing front forks according to a customer’s requirements. Screenshot is taken from PC while PLC is turned off, hence no values.

Figure 28. Temperature control screen.

Figure 29. Cooling control screen.

Figure 30. Velocity control screen.
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>SI unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>[N/m²]</td>
<td>Damping coefficient of the damper</td>
</tr>
<tr>
<td>$c_2$</td>
<td>[N/m²]</td>
<td>Damping coefficient of the tire</td>
</tr>
<tr>
<td>$d_{\text{Gear Cam Drum}}$</td>
<td>[m]</td>
<td>Diameter of Cam drum driveshaft</td>
</tr>
<tr>
<td>$d_{\text{Gear motor}}$</td>
<td>[m]</td>
<td>Diameter of motor driveshaft</td>
</tr>
<tr>
<td>$f_{\text{Supply}}$</td>
<td>[Hz]</td>
<td>Power supply frequency</td>
</tr>
<tr>
<td>$F_1$</td>
<td>[N]</td>
<td>The force affecting the sprung mass</td>
</tr>
<tr>
<td>$F_{1^*}$</td>
<td>[N]</td>
<td>The combined force of the spring and damper on the top spring and damper mount</td>
</tr>
<tr>
<td>$F_2$</td>
<td>[N]</td>
<td>The force affecting the unsprung mass</td>
</tr>
<tr>
<td>$F_{2^*}$</td>
<td>[N]</td>
<td>The total tire force acting on the unsprung mass</td>
</tr>
<tr>
<td>$F_{2^{**}}$</td>
<td>[N]</td>
<td>The combined force of the spring and damper on the bottom spring and damper mount</td>
</tr>
<tr>
<td>$k_1$</td>
<td>[N/m]</td>
<td>Spring stiffness of the spring</td>
</tr>
<tr>
<td>$k_2$</td>
<td>[N/m]</td>
<td>Spring stiffness of the tire</td>
</tr>
<tr>
<td>$L_a$</td>
<td>[m]</td>
<td>Length from swingarm attachment point to wheel centre</td>
</tr>
<tr>
<td>$L_b$</td>
<td>[m]</td>
<td>Length from the swingarm attachment point to lower spring and damper attachment point</td>
</tr>
<tr>
<td>$L_s$</td>
<td>[m]</td>
<td>Length between lower and upper attachment point of swingarm spring and damper</td>
</tr>
<tr>
<td>$l_{TC}$</td>
<td>[m]</td>
<td>Length used to calculate tire compression</td>
</tr>
<tr>
<td>$l_1$</td>
<td>[m]</td>
<td>Length of the rotating beam (see Figure 5)</td>
</tr>
<tr>
<td>$l_2$</td>
<td>[m]</td>
<td>Length from the assumed CoG of the sprung mass to the attachment point of the frame</td>
</tr>
<tr>
<td>$l_3$</td>
<td>[m]</td>
<td>In front suspension model (see Figure 5): Length of the front fork at steady state compression i.e. when the structure is standing still. In rear suspension model (see Figure 6): Length from swingarm attachment point to wheel centre along the swingarm</td>
</tr>
<tr>
<td>$l_4$</td>
<td>[m]</td>
<td>Length from the swingarm attachment point to lower damper attachment point along the swingarm</td>
</tr>
<tr>
<td>$l_5$</td>
<td>[m]</td>
<td>Length from rotating beam attachment point to upper damper attachment point</td>
</tr>
<tr>
<td>$l_6$</td>
<td>[m]</td>
<td>Length from rotating beam attachment point to swingarm attachment point’s along the rotating beam</td>
</tr>
<tr>
<td>$l_7$</td>
<td>[m]</td>
<td>Length from rotating beam attachment point to swingarm attachment point’s along a line tangential to the rotating beam</td>
</tr>
<tr>
<td>Symbol</td>
<td>Unit</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>(l_s)</td>
<td>[m]</td>
<td>Length from swingarm centre line to damper's lower attachment point</td>
</tr>
<tr>
<td>(l_o)</td>
<td>[m]</td>
<td>Length from swingarm centre line to wheel center</td>
</tr>
<tr>
<td>(m_1)</td>
<td>[kg]</td>
<td>Sprung mass of the structure</td>
</tr>
<tr>
<td>(m_2)</td>
<td>[kg]</td>
<td>Unsprung mass of the structure</td>
</tr>
<tr>
<td>(n)</td>
<td></td>
<td>Indicates the current iteration of the simulation process</td>
</tr>
<tr>
<td>(r_{\text{tyre}})</td>
<td>[m]</td>
<td>Tire radius</td>
</tr>
<tr>
<td>(r_{\text{Wheel}})</td>
<td>[m]</td>
<td>Wheel radius</td>
</tr>
<tr>
<td>(w)</td>
<td>[m]</td>
<td>Road profile</td>
</tr>
<tr>
<td>(x_1)</td>
<td>[m]</td>
<td>Displacement of the sprung mass around the attachment point of the frame</td>
</tr>
<tr>
<td>(x_2)</td>
<td>[m]</td>
<td>Displacement of the unsprung mass around the attachment point of the frame</td>
</tr>
<tr>
<td>(x_1^*)</td>
<td>[m]</td>
<td>Displacement of the top of the suspension</td>
</tr>
<tr>
<td>(x_2^*)</td>
<td>[m]</td>
<td>Vertical movement of the unsprung mass</td>
</tr>
<tr>
<td>(x_2^{**})</td>
<td>[m]</td>
<td>Displacement of the bottom of the suspension</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>[rad]</td>
<td>Angle between the rotating beam and the horizontal axis</td>
</tr>
<tr>
<td>(\alpha_0)</td>
<td>[rad]</td>
<td>Angle between the rotating beam and the horizontal axis at steady state</td>
</tr>
<tr>
<td>(\beta)</td>
<td>[rad]</td>
<td>Angle between the sprung mass' direction of travel and the vertical axis</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>[rad]</td>
<td>The angle between the main rotating beam and a line between the rotating beam attachment point and the swingarm attachment point</td>
</tr>
<tr>
<td>(\zeta)</td>
<td>[rad]</td>
<td>Angle used to calculate area of compressed air in the tire cross section</td>
</tr>
<tr>
<td>(\theta)</td>
<td>[rad]</td>
<td>The angle between the swingarm centre line and a line from the swingarm attachment point to the frame and the lower spring and damper attachment point</td>
</tr>
<tr>
<td>(\tau)</td>
<td>[rad]</td>
<td>Vector of angles used to calculate the total area of compressed air in the tire model</td>
</tr>
<tr>
<td>(\phi)</td>
<td>[rad]</td>
<td>The angle between the swingarm centre line and the horizontal axis</td>
</tr>
<tr>
<td>(\chi)</td>
<td>[rad]</td>
<td>The angle between the swingarm centre line and a line from the swingarm attachment point to the frame and the wheel centre</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>Variable Frequency Drive</td>
<td></td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
<td></td>
</tr>
<tr>
<td>CoG</td>
<td>Center of Gravity</td>
<td></td>
</tr>
<tr>
<td>DoF</td>
<td>Degrees of Freedom</td>
<td></td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
<td></td>
</tr>
<tr>
<td>HSC</td>
<td>High Speed Counter</td>
<td></td>
</tr>
<tr>
<td>LAD</td>
<td>Ladder Diagram</td>
<td></td>
</tr>
<tr>
<td>FBD</td>
<td>Function Block Diagram</td>
<td></td>
</tr>
<tr>
<td>SCL</td>
<td>Structured text</td>
<td></td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
<td></td>
</tr>
</tbody>
</table>
## Contents

1  Introduction........................................................................................................................................ 1
   1.1  Background................................................................................................................................. 1
   1.2  Problem definition ..................................................................................................................... 1
   1.3  Outline of thesis ......................................................................................................................... 2

2  Modelling ........................................................................................................................................ 3
   2.1  Preparatory work ....................................................................................................................... 3
   2.2  Modelling theory ....................................................................................................................... 3
      2.2.1  Front suspension geometry ............................................................................................... 4
      2.2.2  Rear suspension geometry ............................................................................................... 6
   2.3  Spring characteristics ................................................................................................................. 10
   2.4  Damper characteristics ............................................................................................................. 11
   2.5  Tire characteristics ................................................................................................................... 11
   2.6  Tire properties during wheel lift ............................................................................................... 15
   2.7  Model validation ....................................................................................................................... 15
      2.7.1  Front suspension validation .............................................................................................. 16
      2.7.2  Rear suspension validation .............................................................................................. 21

3  Digitization of the Cam Drum ............................................................................................................ 22
   3.1  Preparatory work ....................................................................................................................... 22
   3.2  Desired functions ....................................................................................................................... 22
   3.3  Hardware .................................................................................................................................... 23
   3.4  Programming ............................................................................................................................. 23
   3.5  Program layout ......................................................................................................................... 24
   3.6  HMI Program result .................................................................................................................. 26

4  Conclusions ......................................................................................................................................... 30

References .............................................................................................................................................. 31
1 Introduction

1.1 Background
This master thesis was done in collaboration with Öhlins Racing AB. Öhlins Racing was founded in 1976 by Kent Öhlin and began as a producer of shock absorbers for the motocross industry. Over the years the company started manufacturing and developing suspension technology for motorcycle, bicycle and automotive applications in the aftermarket, original equipment manufacturer (OEM) and motorsport industry etc. [1]. Examples of rear suspension systems of a mountain bike and motorcycle are shown in Figure 1 and Figure 2 respectively.

Figure 1. Example of a mountain bike rear suspension layout [2].

One of the many test benches used in the development of new suspension components at Öhlins is the Cam Drum. The Cam Drum is used to simulate a life cycle of dynamic stresses to assess the durability of an assembled suspension system as a whole. Cams of different shapes and sizes are used to simulate road profiles. The Cam Drum that is currently being used at Öhlins Racing AB is driven by an electric AC motor that is controlled by a variable frequency drive (VFD) with a manually adjustable frequency control.

1.2 Problem definition
Öhlins Racing has decided that they would like to digitize the control of the Cam Drum to run more advanced test cycles. By automizing the control of the motor, a predefined test cycle where a set of rotational speeds are maintained for a specific number of rotations each can for instance be accomplished. It is also possible to send data in real time to other computer stations within the facility. This is very beneficial as the Cam Drum is located on the opposite side of the locale from the laboratory.
While the Cam Drum is a useful tool to test suspension components it does require a produced and assembled suspension system before testing can commence. Öhlins has realised that it would be interesting to test conceptual designs prior to production. To do this a computer model of the Cam Drum would be required. So, there are two goals with this thesis. Firstly, to construct a validated suspension model that can be used to correctly simulate the front and rear suspension of a motorcycle and a mountain bike to a degree where the behaviour of a suspension can be estimated prior to being assembled and secondly, to digitize the control of the Cam Drum to allow for more advanced tests and facilitate use of the machine, thus reducing the resources necessary to operate the machine.

### 1.3 Outline of thesis

As previously mentioned, this thesis is split in two parts. Chapter 2 consists of the suspension modelling while digitization of the Cam Drum is covered in Chapter 3. Each chapter is introduced with the preparatory work done before the project was properly initiated. Overall Chapter 2 is largely split in subchapters which deal with the different components of the models while the modelling theory and validation work is divided in two parts i.e. front suspension and rear suspension. Chapter 3 does not contain a lot of programming theory as this would take up far too much space. Focus is instead put on describing the desired functionalities and showing the final result. As this report is intended to be used by engineers at Öhlins for future work with the program, it does go through the layout of the final program.
2 Modelling

2.1 Preparatory work

The Cam Drum is designed to test either the front or rear suspension of a bicycle or motorcycle. The model would therefore be based on a conventional mass-spring-damper system [4] as it is a good model for simulating single suspension systems. Due to the complexity of motorcycle’s and bicycle’s suspension geometries some simplifications need to be made which will be further discussed throughout Chapter 2. The model is made in Matlab, mainly due to the simplicity of the software but also because it is a well-known software among Öhlins Racing employees. Since the model is intended to be used by the people at Öhlins once it is done it is important that there is enough knowledge about the chosen software in case changes are required or errors are found in the future.

Due to the vastly different geometries of the front and rear suspension of bicycles and motorcycles, where the front suspension allows the wheel to move along the front fork as oppose to a rear swingarm suspension which allows the wheel to rotate about the attachment point of the swing arm, two different models were made. Since the Matlab models would share a lot of structural similarities a front suspension model was first constructed and then altered as a rear suspension. This is further discussed in Chapter 2.

2.2 Modelling theory

In a conventional mass-spring-damper system, shown in Figure 3, the movement of the sprung and unsprung mass can be described by the equations of motion as shown in Equations 1.1 and 1.2 which are derived from Newton's second law of motion [5]. One important thing to note here is that the gravitational forces are not included. The displacements are therefore measured from a steady state position i.e. from the position where the compressed springs produce just enough force to support the weight of the structure above that spring. The effect of this will become apparent in later theoretical discussions.

\[
\begin{align*}
    m_1\ddot{x}_1 &= k_1 (x_2 - x_1) + c_1 (\dot{x}_2 - \dot{x}_1) \\
    m_2\ddot{x}_2 &= k_2 (x_1 - x_2) + c_2 (\dot{x}_1 - \dot{x}_2) + k_r (w - x_2)
\end{align*}
\]

The two masses are here constrained to only move in the x- direction. The compressions of the first and second suspension are furthermore calculated as the difference between the movement of the two masses and between the second mass and the road profile respectively. This poses a problem in more complex suspensions, such as the front and rear suspension of a bicycle or motorcycle as shown in Figure 5 and Figure 6, as such simplifications will drastically reduce the validity of the models as the compression and rebound of the springs and dampers can no longer correctly be described by the movement of the masses. To compensate for this, four more displacements are introduced to represent the movement of the top and bottom end of each spring and damper. This is shown in Figure 4.
Equations 1.1 and 1.2 are thereby rewritten as Equations 1.3 and 1.4 with the new displacement variables.

\[ m_1 \ddot{x}_1 = k_1 (x_2^{**} - x_1^*) + c_1 (\dot{x}_2^{***} - \dot{x}_1^*) \] (1.3)

\[ m_2 \ddot{x}_2 = k_1 (x_1^* - x_2^{**}) + k_2 (w^* - x_2^*) + c_1 (\dot{x}_1^* - \dot{x}_2^{**}) + c_2 (w^* - \dot{x}_2^*) \] (1.4)

The relationship between the introduced displacements and conventional displacements depend upon angles varying in time. These will therefore have to be calculated continuously throughout the simulation procedure. The method used to calculate these angles is discussed in Chapters 2.2.1 and 2.2.2.

### 2.2.1 Front suspension geometry

In the case with the front suspension, see Figure 5, angles \( \alpha \) and \( \beta \) are varying with the movement of the structure and are therefore calculated continuously throughout the simulation with Equations 1.5 and 1.6.

\[ \alpha_n = \sin^{-1} \left( \sin(\alpha_o) + \frac{x_1 \cos(\alpha_{n-1})}{l_2} \right) \] (1.5)

\[ \beta_n = \tan^{-1} \left( \frac{l_3}{l_4} \right) - \alpha_n \] (1.6)
\( \alpha_n \) and \( \beta_n \) represents the angles at the current iteration of the simulation and \( \alpha_{n-1} \) represents the angle \( \alpha \) in the last iteration. This is a simplification done to make the calculation easier.

\[ \alpha_{n-1} \]

**Figure 5.** Illustration of the Cam Drum setup of a front suspension with relevant geometrical variables.

It should be noted that the direction of displacement \( x_1 \) and \( x_2 \) in Figure 5 are perpendicular to a line between their respective origin and the attachment point of the frame to the structure of the Cam Drum.

Now that the angles can be calculated the introduced displacement variables can be expressed as functions of the mass displacements according to Table 1.

<table>
<thead>
<tr>
<th>Table 1. Relationship between mass displacements and hardpoint displacements for the front suspension model.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Front suspension</strong></td>
</tr>
<tr>
<td>Hardpoint displacement</td>
</tr>
<tr>
<td>( x_1^* )</td>
</tr>
<tr>
<td>( x_2^* )</td>
</tr>
<tr>
<td>( x_2^* )</td>
</tr>
<tr>
<td>( w^* )</td>
</tr>
</tbody>
</table>
The force produced by the spring and damper will also be affected by the geometry. Table 2 shows how a force produced at the new hardpoints affect the masses.

Table 2. The effect that a force produced at a hardpoint has on one of the masses in their respective CoG (shown as the origin of arrows $x_1$ and $x_2$).

<table>
<thead>
<tr>
<th>Front suspension</th>
<th>Force at hardpoint</th>
<th>Force at mass CoG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F^*_1 \frac{L_1}{L_2}$</td>
<td>$F_1$</td>
<td></td>
</tr>
<tr>
<td>$F^*_2 \cos(\beta)$</td>
<td>$F_2$</td>
<td></td>
</tr>
<tr>
<td>$F^*_2 \cos(\beta - \alpha)$</td>
<td>$F_2$</td>
<td></td>
</tr>
</tbody>
</table>

2.2.2 Rear suspension geometry
The rear suspension geometry is illustrated in Figure 6. Angles $\theta$, $\chi$ and $\gamma$ are static and are calculated as

$$\chi = \tan^{-1}\left( \frac{l_2}{l_1} \right)$$

$$\theta = \tan^{-1}\left( \frac{l_2}{l_4} \right)$$

$$\gamma = \tan^{-1}\left( \frac{l_2}{l_6} \right)$$

Angles $\alpha$, $\beta$ and $\phi$ are varying in time and will have to be calculated continuously throughout the simulation. Due to some similarities in the geometry of the front and rear suspension setup Equation 1.5 can be used to calculate $\alpha$ here as well. To find the equations for angles $\beta$ and $\phi$ a closer analysis of the swingarm and coilover is necessary.
Figure 6. Illustration of the Cam Drum setup of a rear suspension with relevant geometrical variables.

Figure 7 gives a closer look at the geometrical layout of the swingarm with points $a$, $b$ and $c$ being the most important hardpoints of the subsystem. The angle between the line from point $c$ to point $a$ and the horizontal plane is equal to $\phi - \chi$. This relationship will be used to calculate $\phi$.

Figure 7. Geometrical illustration of the rear suspension swingarm.
Based on Figure 8 the following relationship can be set up

\[ l_a \sin(\phi - \chi) = l_a \sin(\phi_0 - \chi) + \Delta_{A,y} - \Delta_{C,y} \quad (1.7) \]

where \( \Delta_{A,y} \) and \( \Delta_{C,y} \) (point a’s and c’s vertical displacement from their original position) can be expressed as

\[ \Delta_{A,y} = x_2 \cos(\phi_{n-1} - \chi) \quad (1.8) \]
\[ \Delta_{C,y} = l_x \left( \cos(\gamma + \alpha_0) - \cos(\gamma + \alpha) \right) \quad (1.9) \]

where \( \phi_{n-1} \) denotes \( \phi \) from the last iteration. This is, just like in the case with \( \alpha \), a simplification made to simplify the calculation.

Combining Equations 1.7, 1.8 and 1.9 gives us

\[ \phi = \sin^{-1} \left( \frac{l_a \sin(\phi_0 - \chi) + x_2 \cos(\phi_{n-1} - \chi) + l_x \left( \cos(\gamma + \alpha_0) - \cos(\gamma + \alpha) \right)}{l_a} \right) + \chi \quad (1.10) \]

Because the rear coilover on its own has 5 DoF the compression/rebound will be split into an \( x \) and \( y \) part. The total compression/rebound will then be calculated using Pythagoras’ theorem. An illustration of this is shown in Figure 9.
From Figure 9 it can be seen that $\beta$ can be acquired as

$$\beta = \tan^{-1}\left(\frac{A}{B}\right)$$  \hspace{1cm} (1.11)

Lengths $A$ and $B$ in turn can be calculated as

$$A = A_0 + \dot{x}_{1,x} - \dot{x}_{2,x} = l_{S,0} \sin(\beta_0) + \dot{x}_{1,x} - \dot{x}_{2,x}$$  \hspace{1cm} (1.12)

and

$$B = B_0 + \dot{x}_{1,y} - \dot{x}_{2,y} = l_{S,0} \cos(\beta_0) + \dot{x}_{1,y} - \dot{x}_{2,y}$$  \hspace{1cm} (1.13)

Now combining Equation 1.11 with Equations 1.12 and 1.13 gives

$$\beta = \tan^{-1}\left(\frac{l_{S,0} \sin(\beta_0) + \dot{x}_{1,x} - \dot{x}_{2,x}}{l_{S,0} \cos(\beta_0) + \dot{x}_{1,y} - \dot{x}_{2,y}}\right)$$  \hspace{1cm} (1.14)

The compression/rebound of the spring is calculated according to Equation 1.15

$$\Delta_s = l_s - l_{S,0} = \sqrt{A^2 + B^2} - l_{S,0}$$  \hspace{1cm} (1.15)

The expressions $x_1 - x_2$ and $\dot{x}_1 - \dot{x}_2$ in Equations 1.3 and 1.4 will therefore be replaced by $\Delta_s$ and $\dot{\Delta}_s$ in the case of the rear suspension.

Table 3 shows the relationship between the mass displacements and the introduced displacements for the rear suspension. As the compression of the tire caused by the road
profile is equal to the displacement of the road profile in both suspension variants the introduced variable $w^*$ will be referred to as $w$ from here on.

Table 3. Relationship between mass displacements and hardpoint displacements for the rear suspension model

<table>
<thead>
<tr>
<th>Hardpoint displacement</th>
<th>Mass displacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^* - x_2^*$</td>
<td>$\sqrt{(l_{5.0} \sin(\beta_0) + x_{1,x}^* - x_{2,x}^<em>)^2 + (l_{5.0} \cos(\beta_0) + x_{1,y}^</em> - x_{2,y}^*)^2}$</td>
</tr>
<tr>
<td>$x_2^*$</td>
<td>$x_2 \cos(\phi - \chi)$</td>
</tr>
<tr>
<td>$w^*$</td>
<td>$w$</td>
</tr>
</tbody>
</table>

Once again, the transmission of forces produced by the coilover will be affected by the geometry. The conversions are shown in Table 4.

Table 4. The effect that a force produced at a hardpoint has on one of the masses in their respective CoG (shown as the origin of arrows $x_1$ and $x_2$).

<table>
<thead>
<tr>
<th>Rear suspension</th>
<th>Force at hardpoint</th>
<th>Force at mass CoG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_1^* \frac{L_2}{L_1} \cos(\alpha - \beta)$</td>
<td>$F_1$</td>
</tr>
<tr>
<td></td>
<td>$F_1^* \frac{L_1}{L_2} \cos(\beta + \chi - \phi)$</td>
<td>$F_2$</td>
</tr>
<tr>
<td></td>
<td>$F_2^* \cos(\phi - \chi)$</td>
<td>$F_2$</td>
</tr>
</tbody>
</table>

2.3 Spring characteristics

As a part of the development and testing processes the springs are tested using dynamometers to log the spring characteristics. These are generally represented in graphs by plotting force as a function of compression or rebound. However, as the intended area of use for the model is as a tool in the development process exact spring curves might not always be accessible. The user will therefore be given the choice between a linear and a non-linear spring model.

The linear spring model consists of a gain block with the spring stiffness value. This gain is then multiplied with the compression/rebound to get the force produced by the spring. The non-linear spring model uses vectors for compression and rebound and the corresponding force vectors. The sign of the spring displacement velocities is used to assess whether the spring is in a compression or rebound situation and the corresponding vectors are used to look up the force equivalent to the spring’s current length. Whether the spring is in a compression or rebound situation has no effect on the spring itself but rather on the impact rubber. However, since the impact rubber is included in the spring rate-curve the spring model has to take this into account.

Due to the earlier choice to exclude the gravitational force’s impact on the system, the force corresponding to zero displacement of the spring will in practice correspond to the steady state displacement while standing still. Thus, each point in the spring displacement vector is
subtracted by an estimated initial compression of the spring. If the suspension can be assembled this value can be measured. If, however, a non-existing suspension is to be simulated this value has to be estimated.

2.4 Damper characteristics
The user is once again given the choice to choose between a linear and a non-linear model. The linear damper model multiplies the damper velocity with one of two linear damping coefficients depending on the situation, namely compression or rebound, to get the force produced by the damper.

For the non-linear damper model, roughly 36 forces are read from a standardised excel-sheet given by the damper dynamometer software. These forces signify the force produced by the damper at 36 different compression/rebound velocities ranging from -1 to 1 where negative velocities represent rebound. Interpolation is then used between these points to calculate the force equivalent to the compression/rebound velocity, i.e. the equivalent damping coefficient, throughout the simulation. For velocities outside of this range a linear force to velocity ratio is assumed.

2.5 Tire characteristics
One of the major drawbacks of the conventional mass-spring-damper system is that the compression of the tire is expressed using the vertical offset of the tire and road profile. Only the tire’s centreline will therefore be considered when calculating the compression of the tire. Figure 10 illustrates a situation where this can be problematic. The bump has almost moved halfway past the tire yet since the middle of the tire is still in contact with flat ground no compression has been registered. This had a clear effect on early comparisons between logged validation signals and simulation results.

![Figure 10. Illustration of how using two points to calculate tire forces can be problematic.](image)

To compensate for this, a tire model was reconstructed to make the forces produced by the tire proportional to the amount of compressed air and not the difference between two points. This was done by splitting the bottom half of the tire into 180 cross sections corresponding to the angle $\tau$ in Figure 11 taking on all angles from -90 to 90 degrees. The length $l_{TC}$ represents the length from the centre of the tire to the road profile following a given angle. This length can be used to calculate the compression of the tire along a line in the direction of an angle $\tau$ according to
\[ \delta_T = r_{\text{wheel}} - l_{TC} \quad (1.16) \]

where \( r_{\text{wheel}} \) is the radius of the unloaded wheel.

In order to calculate \( l_{TC} \) for each \( \tau \) a local coordinate system is introduced where the bottom centre point of the wheel is origo. The height of the road profile at flat ground is moved up from the \( x \) axis by an estimated deformation of the tire at equilibrium, i.e. steady state of the loaded tire, acquired with linear tire model theory through Equation 1.17. This is done to account of the initial deformation of the tire from the weight of the structure as a time period for the tire to find its steady state would otherwise have to be implemented before each simulation. In other words, each simulation starts with a compression of the tire corresponding to the deformation at steady state.

\[
\delta_{\text{Equilibrium}} = \frac{N_{\text{Equilibrium}}}{k_2} \quad (1.17)
\]

where the normal force at equilibrium is calculated as

\[
N_{\text{Initial}} = \frac{m_1 g l_2 \cos(\alpha)}{\sqrt{l_3^2 + l_4^2 \cos\left(\tan^{-1}\left(\frac{l_3}{l_4}\right) - \alpha\right)}} + m_2 g \quad (1.18)
\]

for the front suspension and

\[
N_{\text{Initial}} = \frac{m_1 g l_2 \cos(\alpha)}{\cos\left(\frac{\pi}{2} - \gamma - \alpha\right) L_x + \cos(\phi) L_3} + m_2 g \quad (1.19)
\]

for the rear suspension. With this initial compression the tire model will initially produce some force. This is solved by calculating the initial compressed volume of air prior to simulation initiation and removing this from the calculated volume in the tire model throughout the simulation. By doing this the tire is allowed to produce a negative force equal to the normal force of the tire at equilibrium while the Cam Drum is standing still. This will be explained further in Chapter 2.6.

Based on this new coordinate system the following equations can be set up

\[ x = l_{TC} \sin(\tau) \quad (1.20) \]

\[ y = l_{TC} \left(1 - \cos(\tau)\right) \quad (1.21) \]

where \( x \) and \( y \) represent the road profile’s \( x \) and \( y \) coordinate in the new coordinate system. Rewriting Equations 1.20 and 1.21 gives Equation 1.22.
\[ \frac{x}{r_{\text{Wheel}} - y} = \tan(\tau) \]  

(1.22)

With Equation 1.22 the tire model finds the road profile data corresponding to each angle \( \tau \) from -90 to 90 degrees. Equation 1.20 is then used to calculate \( l_{TC} \) and Equation 1.16 is finally used to calculate \( \delta_{T} \) for each angle \( \tau \). \( \delta_{T} \) can of course not be smaller than zero or larger than \( r_{\text{Tire}} \). Each \( \delta_{T} \) is then multiplied by \( \frac{2(r_{\text{Wheel}} - r_{\text{Tire}})\pi}{2 \times 180} \), which is roughly equivalent to the length between each line, before being summed up as the total area of compressed air.

Following is a suggestion on how the model can potentially be further developed to depend on the total volume of compressed air. This was tested in this model but was found to give very small differences compared to area dependent characteristics and was therefore not used in the end. If a very high accuracy is required, this might be useful although it is simply unproven theory so far.

![Figure 11. Lateral view of the wheel and road profile.](image)

Looking at the tire cross sections, see Figure 12, \( \delta_{T} \) is now known. However, there are still one unknown variables of interest needed to calculate the volume of compressed air, namely \( \zeta \). The angle \( \zeta \) is calculated according to

\[ \zeta = \cos^{-1}\left(\frac{r_{\text{Tire}} - \delta_{T}}{r_{\text{Tire}}}\right) \]  

(1.23)

The bottom area in Figure 12 is now calculated as

\[ A_{\text{Compressed air}} = r_{\text{Tire}}^2 \pi \frac{2\zeta}{2\pi} - r_{\text{Tire}} \sin(\zeta) r_{\text{Tire}} \cos(\zeta) \]  

(1.24)
The total volume of compressed air is finally calculated by summing up all individually calculated areas of compressed air and multiplying them by \( \frac{2(r_{\text{wheel}} - r_{\text{Tire}})\pi}{2*180} \).

![Figure 12. Cross section of the tire.](image)

At a point during the validation work it became apparent that something was wrong with the model theory. It turned out that this was most likely due to the simplifications done with regards to the movement of the unsprung mass. When describing the movement of the unsprung mass with regards to the ground the movement caused by the upper mass that is not transferred through the suspension is disregarded. To solve this the tire model was altered by moving the road profile in the local coordinate system vertically a length

\[
\ell = l_\beta \left( \sin(\alpha_\theta - \beta) - \sin(\alpha - \beta) \right) = l_1 - l_2
\]

(1.25)

where lengths \( l_1 \) and \( l_2 \) in this case refers to the lengths shown in Figure 13. The effect of this is shown in Chapter 2.7.1.

![Figure 13. Cam drum fork geometry in steady state (solid lines) and after hitting a bump (dashed lines).](image)
2.6 Tire properties during wheel lift
In real life a tire is never able to provide a pulling force towards the ground. However, since the tire force produced when the system is in equilibrium is considered to be zero, a negative force equal to the initial normal force can be achieved in this system. When this force is reached, the tire is in reality producing no force. The negative force simply corresponds to the gravitational force. The ability to produce negative force is achieved by including the initial compression of the tire as discussed in Chapter 2.5 in the tire model.

Figure 14 shows the model logic of going from normal operations to a wheel lift scenario. When “F_däck” i.e. the normal force at the tire reaches a negative value equal to the initial normal force switch 1 changes the output from the calculated normal force of the tire to the initial normal force multiplied by -1. As a precautionary measure, a second switch is included. When the tire has completely left the ground, the initial normal force multiplied by -1 is once again fed through.

![Figure 14. Wheel lift logic of the model.](image)

2.7 Model validation
To validate the models, validation signals from actual Cam Drum test runs were required. What the optimal type of signal is for this purpose is not a simple question to answer. According to Newton’s second law of motion it is the acceleration as the forces in the system will be a direct result of the accelerations of the various parts. For spring and damper analysis purposes one could argue that position and velocity is more important as these parameters determine the force exerted from the spring and damper. Position and velocity signals have therefore primarily been used throughout the validation.

At an early stage of the project no sensors were available. A software called Tracker [6] was therefore used to manually extract position signals from slow motion videos of various Cam Drum runs. Videos were captured using a Samsung S7 mobile phone able to capture video at 60 Hz [7]. This made it possible to track points on the sprung and unsprung masses. At a later stage a GET M40 position sensor from manufacturer Athena became available. The data acquired with Tracker gives a good understanding of the differences between simulated positions and acquired positions as the position of the masses are shown individually while data acquired with the position sensor gives a more precise comparison between the behaviour of the springs and dampers as the position sensor data shows how much the spring and damper length has changed from steady state. Both systems were therefore finally used in the validation work.
Two front forks and one rear swingarm were in total used to gather validation data. Both front forks were mountain bike type forks and the rear swingarm was a street motorcycle swingarm. Öhlins facility in Upplands Väsby does not test mountain bike swingarms and rarely tests motorcycle front forks. No recordings of these types were therefore acquired. The advantage of having at least one motorcycle and one mountain bike test object was that the tire model could be validated for both types of tires.

### 2.7.1 Front suspension validation

The first recording session was done at 12 km/h with one 50 mm bump and two 30 mm bumps using fork #1. At this time no position sensor was available. Data was therefore gathered using the Tracker software solely. Figure 15 shows a comparison between the recorded and simulated positions of the sprung and unsprung masses after some validation work. The validation was here focused on finding good tire parameters (tire stiffness and damping coefficient) for the new tire model. Albeit with some deviations from the actual signal, mainly around the smaller bumps, the model was deemed good enough to move on with.

![Comparison between simulation and camera recorded positions](image)

**Figure 15.** Comparison between recorded Tracker data from the first data recording on front fork #1 at 12 km/h (tangential speed) and model simulation positions after some validation work.

A separate damper and spring combination was simulated and recorded (Front fork #2), this time using Tracker and the position sensor. Figure 16 shows a comparison between the model simulation versus recorded Tracker data and Figure 17 shows a comparison between the position sensor data and the model compression and rebound.
Figure 16. Comparison between recorded Tracker data from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation positions.

Figure 17. Comparison between recorded position sensor data (difference in spring and damper length from steady state) from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation compression and rebound.

By analysing Figure 15 and Figure 16 it can clearly be seen that there are large differences in the behaviour of the model and the recorded data. This indicates that there is a fundamental error in the model. The cause of this error and the measures taken to resolve this is discussed in Chapter 2.5. Figure 18 and Figure 19 shows comparisons between recorded Tracker and position data and the simulated model data with the altered tire model.
By analysing Figure 19 it can be seen that there are clear improvements in the result as compared to Figure 17 in terms of behaviour. The graphs show similar tendencies and follow each other quite well overall. There are still some overestimations on the compression side and the rebound behaviour of the model shows two peaks while the position sensor shows one. Figure 18 does however show significantly worse results for the unsprung mass while the sprung mass shows a slightly better correlation. This is interpreted as a result of the way the tire model was altered. At this point it was decided that if a good correlation could be found between the position sensor data and model results for several models the deviations in Tracker would be disregarded as the field of interest is the behaviour of the spring and damper. Figure 20 shows the derivative of the signals from Figure 19 i.e. the velocities of the
spring and damper from the position sensor and model simulation. Once again there is a clear resemblance between the two signals.

Figure 20. Comparison between recorded spring and damper velocity from the second data recording on front fork #2 at 12 km/h (tangential speed) and model simulation compression and rebound velocity using the updated tire model. Both signals filtered using a Butterworth filter.

This model was compared to a few other test signals, one of the more extreme ones being a run at a tangential speed of 45 km/h with one 50 mm bump. Figure 21 and Figure 22 shows a comparison between the recorded and simulated signal data and their derivative respectively.
Figure 21. Comparison between recorded position sensor data (difference in spring and damper length from steady state) from the second data recording on front fork #2 at 45 km/h (tangential speed) and model simulation compression and rebound using the updated tire model.

Figure 22. Comparison between recorded spring and damper velocity from a data recording on front fork #2 at 45 km/h (tangential speed) and model simulation compression and rebound velocity using the updated tire model. Both signals filtered using a Butterworth filter.

The main difference that can be spotted here is the more significant amount of oscillations after the bumps in the model. The oscillations seem to start as the wheel is compressed against the ground after the bump.
2.7.2 Rear suspension validation

Unfortunately, neither time nor sufficient data was sufficient enough to perform a proper validation of the swingarm model. Only one recording was gathered with Tracker. As discussed in Chapter 2.7.2 it turns out that a position sensor signal is required to properly validate the behaviour of the modelled spring and damper.

Figure 23 shows a comparison between the recorded swingarm data and the simulated positions of the unsprung and sprung mass. This simulation is made after having validated the tire parameters against this data recording.

Figure 23. Comparison between recorded Tracker data from the first data recording on the swingarm at 18 km/h (tangential speed) and model simulation positions after some validation work.


3 Digitization of the Cam Drum

Since the installation of the PLC control will be a vital part of the everyday operations at Öhlins’ laboratory this chapter is partially structured as an instruction manual. Testing programs are regularly changing and new programs are frequently added. It is therefore important that this document provides sufficient information about the layout of the program to allow engineers at Öhlins to alter the program to account for future requirement changes. The focus on program layout might therefore seem overwhelming at times to parties not involved in further developing the program.

3.1 Preparatory work

The control of the VFD was to be handled by a programmable logic controller (PLC) via a suitable fieldbus protocol. One of the first objectives was therefore to investigate different options in terms of hardware supplier, necessary subcomponents and communication protocol. It was quickly decided upon that Siemens would supply the hardware as several work stations around the facility already employ Siemens’ technology.

Siemens’ most basic controller, the Simatic S7-1200 [8], was deemed appropriate for the task at hand as more advanced controllers are more suitable in automated production where several work stations are to be controlled and work in synch.

The current VFD, an NFO Sinus IP20 supplied by NFO Drives, supports several communication protocols [9]. The most common being Modbus RTU and Profibus. The main differences between these communication protocols is that Profibus is very standardized making communication between machines from different manufacturers easier while Modbus RTU can be very individual between manufacturers making cross-manufacturer communication harder. In this case however Modbus RTU was directly supported by the VFD while Profibus required an external communication card. It was therefore decided that Modbus RTU was to be used.

3.2 Desired functions

As previously mentioned the main goal of automizing the PLC controls was to be able to run a program with varying rotational Cam Drum speed. A PLC controlled VFD with an HMI screen interface does however offer other advantages. When a technician at Öhlins was to run a Cam Drum program before the PLC was installed program specific parameters such as rotational speed of the Cam Drum and number of rotations before the program is done had to be set manually. This would occasionally be time consuming as the desired rotational speed was set as a frequency on the VFD meaning a conversion factor between the VFD frequency and Cam Drum frequency would have to be considered. Furthermore, by studying Cam Drum runs performed while this project was carried out it was found that the rotational speed of the Cam Drum would regularly differ from the desired speed by roughly 1%. This would imply that the manual control of the frequency is either prone to some error or not precise enough.

With the PLC controller however, the technician will simply push a button on the HMI screen corresponding to the desired Cam Drum program to set all necessary parameters. Any precision errors related to the manual control will thereby be eliminated as a calibrated decimal value corresponding to the desired rotational speed will be used to specify the desired speed.
Another time-consuming factor was the initiation procedure of the Cam Drum with regards to thermometer control. The thermometer was set to stop the Cam Drum in case the ambient temperature surpasses the pre-specified limits of the working range. An upper and a lower limit is usually set. This feature could however not be enabled until the working temperature had reached the lower limit meaning the technician would have to come back later to do this. With the PLC this restriction was eliminated by automatically enabling the temperature monitoring system when the lower limit plus a hysteresis factor was reached.

If the motor is started without resetting the count value some logic is desired. If the count limit is reached the count value is automatically reset prior to restart. If the count limit is not reached at the time of restart the counter is not reset.

Control of the cooling system is handled by several regulators. Two methods of cooling are available; air stream-cooled and fan cooled. The main regulator controls which of the two systems is active and this is controlled via a switch on the door of the cabinet. In the new application the 24 VDC air stream regulator will be controlled by one of the internal regulators of the PLC. The main regulator will be used to engage and disengage the 230 VAC outlet supplying the electric fan and opening and closing this regulator will be done with an internal PLC-regulator. Control of the internal regulators is discussed in Chapter 3.5. A separate fan is available within the motor casing. This is controlled by yet another regulator which will be controlled by an internal PLC-regulator.

The chosen software allows mathematical equations to be used on parameters. This is utilized in this application to calculate the time remaining until the count limit is reached with the given rotational speed of the program.

### 3.3 Hardware

Siemens’ PLCs comes equipped with many options for sending and receiving digital, analogue and serial signals. It offers several internally integrated features, one such being an high speed counter (HSC). This renders the need for an external counter useless. However, in an attempt to ease a potential transition back to the old system in the event of a major malfunction the counter was kept and reprogrammed to give an output impulse signal per rotation as output. This was then fed to the HSC.

The thermometer previously used was a type K thermocouple. As there are several benefits associated with this type of thermometer (inexpensive, reliable, accurate etc [10]) the same type is to be used for this application. A Simatic Thermocouple Signal Module [11] was therefore implemented.

### 3.4 Programming

The PLC is programmed in Siemens’ own software, SIMATIC Step 7 (TIA Portal), which according to Siemens is the world’s most widely used programming software in industrial automation [12]. The software supports the programming languages LAD (Ladder Diagram), FBD (Function Block Diagram) and SCL (Structured text) [13]. In this application a combination of LAD and FBD is used to construct the PLC program.

As previously mentioned the chosen method of communication between the PLC and VFD is by Modbus RTU serial communication. This serial protocol uses a master-slave based communication method where the slave’s (VFD’s) parameters can be written to and read by the master (PLC). The slave’s parameters are used to control all functions of the VFD which
means that the PLC is able to control the VFD by writing these parameters. The available parameters and their respective functionalities have been chosen by NFO Drives [9] and cannot be altered.

3.5 Program layout
In network 1, see Figure 24, the RS232 communication port used for data transfer between the PLC and VFD is configured. Baud rate and parity is set in accordance with the corresponding values of the VFD (Baud rate: 9600 Bd, Parity: No parity). Input “REQ” executes the block when the input signal is true, in this case when “#Initial call” is true i.e. during the first cycle after start-up.

![Network 1: Configuration of communication port.](image)

Network 2 contains the modbus master-block, see Figure 25. All communication (sending and receiving data) with the VFD is done through this block. Inputs DATA_ADDR and DATA_LEN specifies the desired parameter to read or write and therefore must be changed continuously. Input mode specifies whether the master is to execute a read or write operation and is therefore also changed frequently.
Due to the inherent logic of Modbus communication the modbus master must receive an answer after sending a message before a new message can be sent. Failure to comply to this gives a busy message as a response. The sending and receiving logic comprises networks 3 to 5 and 6 to 14 respectively. In normal operation the PLC cycles through networks 6 to 14, continuously updating read parameters. When a write request is given, either by actively pressing a button on the HMI or through a programmed event, the current read cycle is finalized before executing networks 3 through 5.

Network 16 comprises the write button-commands available in the HMI interface (start, stop, change pre-programmed speed etc.) and the write logic behind the cyclic program earlier mentioned. When a button is pressed the corresponding send address and send data is sent to the modbus master-block and a write request is set. The write logic of the cyclic program automatically executes the same procedure as the button-activated write commands once a given number of counter values are reached. This program uses a separate counter value which is reset every time a cycle is completed.

Network 17 sets a request to send a stop signal once at least one of three types of stop triggers are activated. The bit controlling the motor fan regulator is also reset once a trigger becomes true.

Network 18 reads the current count value recorded by the HSC and creates a reset value by subtracting the total count value with the momentary count value when the reset button is pressed. Likewise, network 19 creates a count value with reset for the cyclic program. This value is reset once a trigger is sent from network 16.
Network 21 sets a stop trigger once the counter value of the active program reaches the set count-limit.

Network 22 sets a 500 millisecond “true” signal once a RUN-button is pressed which sets the corresponding active-bit. This is a safety measure as the bit is set directly with the button but might be reset if the count target is already reached and thus reset after the start button is pressed, thereby stopping the motor. The chance of this happening is greatly reduced by making sure the “set bit” signal is active for 500 milliseconds. The regulator controlling the motor fan is also closed once any start button is pressed.

Network 23 calculates the rotational frequency and tangential speed of the Cam Drum for the individual programs based on their programmed VFD frequency. Each frequency is divided by 16.1, as this corresponds to a rotational frequency of 1 Hz according to Equation (1.26)

\[
f_{CamDrum} = \frac{1}{f_{Supply}} \frac{d_{Gear\ CamDrum}}{d_{Gear\ motor}}
\]

and then multiplied by 12, which is the tangential speed in km/h at a rotational frequency of 1 Hz according to Equation (1.27).

\[
v = \frac{d}{r} = d \pi f
\]

The calculated rotational frequency is then used in network 24 to calculate the remaining time before the count limit is reached. This is simply done by dividing the number of remaining rotations with the rotational frequency to get the remaining time in seconds.

Network 25 reads the value of the four thermometers and creates new parameters with these values. These values are then used in network 26 where the thermometer control is located. This network send a stop trigger once a thermometer reaches a value above the upper limit or below the lower limit of the acceptable range. In order to send this stop trigger the system must have reached the lower limit plus a hysteresis factor of 1 °C as this sets a bit which is required to send the stop request.

Networks 28 and 29 makes up the cooling control. A temperature limit above which the system should activate is set on the HMI. Activating the system and choosing which method of cooling to use is also done on the HMI. The cooling systems can also be tested on the HMI. Network 29 combines the system run signals from network 28 to close the necessary internal relays.

### 3.6 HMI Program result

When the PLC is started the root screen is activated (see Figure 26). From here a connection between the PLC and VFD is first established using the “Initiering”-button which puts the VFD in “Bus standby”. This must always be done prior to any write instructions [9]. The motor control is firstly split up into programs (“Program”) and manual control (“Manuell”). Manual control allows the technician to use the old frequency switch to control motor speed while programs include all pre-programmed programs.
As an example, a customer-specified test program is shown in Figure 27. The layout was designed according to counsels from Öhlin’s laboratory personnel in an attempt to make it as user friendly as possible. At the top is a bar used to navigate through the HMI found on most pages of the HMI. A button for opening an information screen is also available in most pages. On the left is some information regarding the program in question and on the right are two buttons for starting and stopping the motor and a button for opening the cooling control screen. At the bottom is (from the left) a button for opening the temperature control screen, displays showing the recorded temperature at the four thermometers, the desired tangential velocity and rotational frequency in km/h and Hz and a button for opening velocity control, number of rotations since the last reset and the number of rotations at which point the motor will shut down, the reset button and the calculated time until the goal number of rotations is reached.

Figure 27. The PLC test program screen for testing front forks according to a customer’s requirements. Screenshot is taken from PC while PLC is turned off, hence no values.
Figure 28 shows the temperature control screen. The displays under “Undre gräns” and “Övre gräns” are used to set the desired lower and upper limits for the individual thermometers. The buttons under “Aktiv/Inaktiv” are used to activate the thermometers which allows them to turn the motor off if their intervals are breached. The green light under “Intervall” will illuminate once the lower limit plus the hysteresis factor is reached indicating that that thermometer will stop the motor if its limit is breached.

![Temperature control screen](image)

Figure 28. Temperature control screen.

The cooling system control screen is shown in Figure 29. On the far left the limit above which the chosen cooling method engages. The drag switch is used to choose between the two cooling methods. The “På/av”-button is used to activate and deactivate the cooling system. “Test luft” and “Test fläkt” is used to test the air and fan system.

![Cooling control screen](image)

Figure 29. Cooling control screen.

The velocity control screen shown in Figure 30 is used to change the velocities of the individual programs. This was briefly discussed in Chapter 3.4 regarding network 16 where
the logic behind this screen is located. The VFD stores seven frequencies which are, in this application, used to run the programs which use in total 7 different speeds (12, 18, 20, 25, 35, 40 and 45 km/h). If the technician experiences oscillatory behaviour due to a resonance frequency close to the predefined velocity it might be necessary to slightly change the velocity. Rows “Nuvarande värde” shows the current value of the parameters. If the technician wishes to change this value the desired value is written in the row “Ändra” which changes the parameter stored in the PLC. The corresponding “Ändra”-button is then pressed in order to send the new value to the VFD. The frequencies shown in Figure 30 are power supply frequencies i.e. 16.1 Hz here corresponds to a rotational frequency of 1 Hz.

Figure 30. Velocity control screen.
4 Conclusions

The digital motor control system has been live tested with good results. Tangential speeds are found to be within roughly 1/100 km/h from the desired value. Overall response from the technicians has been good and all desired functionalities are now available apart from the ability to receive live data in the laboratory. Due to IT related security reasons, sending webservice data over the buildings industrial network is problematic. Overall, however, the start-up procedure and continuous maintenance of the machine during operation has been greatly facilitated and automated tests are now possible to conduct.

The validated front suspension model shows a good correlation to sampled data implying that the theory on which it is based is valid. Deviations such as under and over estimations and a higher oscillating frequency when settling after a bump induced event are still present. The under and over estimations could potentially be an effect of badly measured input parameters while the higher oscillating frequency is likely an effect of the tire model being underdamped. The model could be used to estimate the behaviour of a front mountain bike suspension but should preferably be tested against another front fork before doing so as the final validation work was based on runs with a single front fork. After the adjustment to the tire model the individual position signals of the sprung and unsprung mass does not show accurate results. It is therefore suggested that the difference in spring and damper length and its derivatives is used to evaluate the results.

The rear suspension model shows a similar correlation with the sampled data to the front suspension after being validated against Tracker data. This implies that the same kind of adjustment done to the tire model in the front suspension could potentially give a good estimation of the spring and damper behaviour. It is still not usable though as this needs to be done prior to any actual application of the model.

The unconventional tire model designed in this project shows good potential however the exact effect of the tire model as compared to a conventional 1-dimensional tire model was not investigated. A future comparison along with validation data would be of great interest if the tire model is to be utilized and developed further.
References


