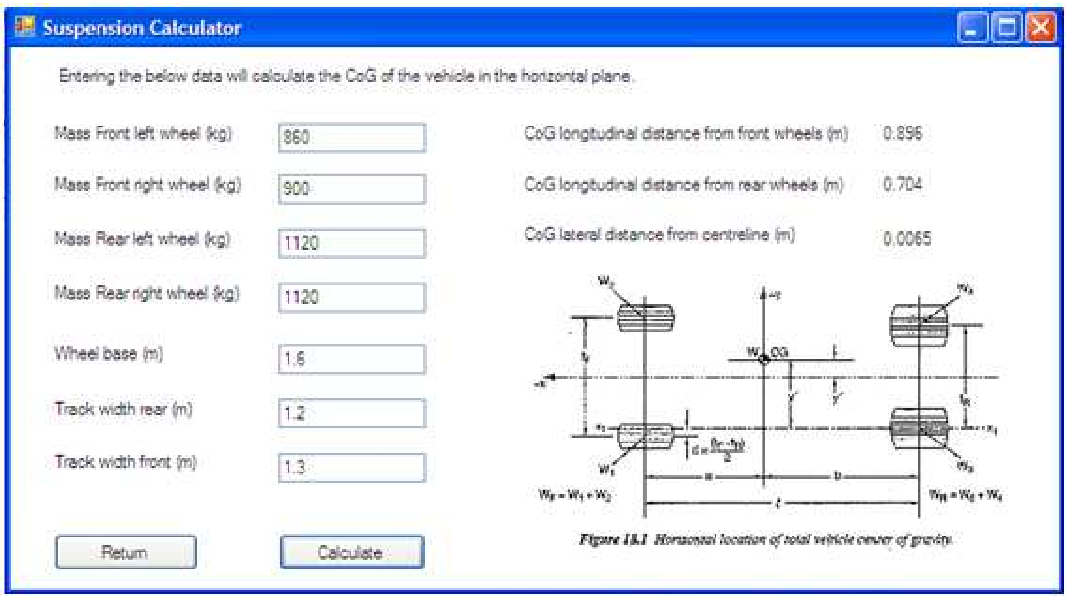
Suspension Calculator

Project Progress Report



Supervisor:
Professor Michael J. S. Lowe

Associate Supervisor:
Dr. Frederic B. Cegla

Cristobal Lowery

Mechanical Engineering
Imperial College London

February 4, 2010

1 Abstract

This report aims to discuss and evaluate the progress made in the suspension calculator project since the Project Plan Report. It begins with a brief introduction to the background, purpose and context of the work. There is then a general discussion of the methods that have been used in programming the suspension calculator. The specific tasks that have been worked on are then discussed individually, according to the initial schedule of work that had been laid down in the Gantt chart. For each task, there is a description of the work that has been performed so far, the problems that have been encountered, and any outstanding work. Progress is then discussed in more holistic terms, where the delays that have been encountered are explained and changes to the remainder of the schedule are justified, with provision of a revised Gantt chart. Finally the report summarises the progress of the project to date. It concludes on the usefulness of the Gantt chart as a planning tool, and, that, given the existence of many unknown elements involved at the initial planning stage, the project has not been unsuccessful in keeping in line with its schedule during the first term of work.

Table of Contents

2 Introduction

This project is an extension of previous work and experience with the Imperial Racing Green organisation, which involved designing, making and testing the suspension system for a zero-emissions fuel cell race car. (1) The principal aim of the project is to create and promote an open-source program for performing suspension calculations using Microsoft Visual C#. (2) The program will also provide a teaching platform with diagrams and written explanations. Further details of the project's aims and functions are outlined in the Project Plan Report. (3)

This report will discuss the progress of the project since the Project Plan Report, in order to evaluate the work that has already been done and to revise the plans for completing the outstanding work. Currently, the majority of work has been on research, learning, and programming the calculating functions of the suspension calculator. Some work has been done on improving its visual appearance and promoting the project through the use of the internet, but these tasks are largely yet to be completed. The report is broken down into discussion of the tasks, timeline, and conclusions. There is individual discussion on how work has been carried out to date, the challenges that have been encountered, and the work that is left to be completed, for each task. Then the originally proposed timeline is compared with the work as it was actually completed. A revision of the timetable for the remaining project is provided, before the final conclusions.

3 Tasks

3.1 Calculations

The project's focus to date has been on the creation of the 'calculations' section of the software, which was completed on January 28, 2010. The calculations section consists of a main window which has 19 'buttons', all of which open up a new window in which the user can perform a specific calculation. As can be seen in Figure 1, the 19 buttons can be arranged into 5 groups; centre of gravity (CoG) buttons, wheel loads buttons, spring buttons, anti-roll bar & roll/ride rates buttons, and wishbones related buttons.

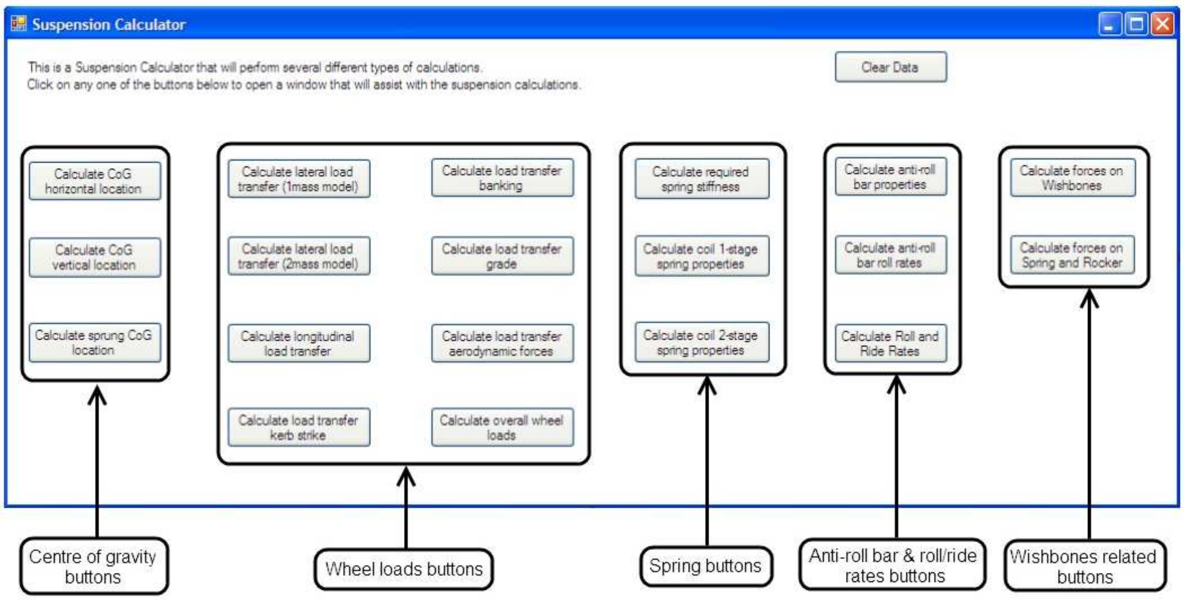


Figure 1: Opening screen of software program

Initially these groupings were designed to reflect the same 4 calculation tasks as set out in the Gantt chart. However, they are now slightly different due to a few changes in the plan. This report is structured and written based on the original tasks found in the Gantt chart, with one additional section for these 'unforeseen' changes which have been encountered whilst working on the project.

To date, the program consists of 5900 lines of self-written code, and 7800 lines of code generated through the graphical user interface (GUI). The programming of calculations for all the sections is now complete. However, at a later stage in the project, some improvements may be added to the wishbone calculations.

The work to complete the calculations section involved several stages. The first was to learn how to use Microsoft Visual C#'s GUI and programming language with the aid of books (4) (5) and online tutorials (6) (7). The next step was to create a user interface for each button and to then program these. The final stage of work in the calculations section was to test what had been programmed by comparing values obtained from the program to ones obtained by performing the same calculations by hand.

More specifically, to program these buttons, it was necessary to gain a deeper knowledge of the underlying engineering theory, to work out which data would be needed and which formulas would be used. The actual programming involved 4 methods, or routines, which were kept consistent for each button as the calculator will be an open source program. The methods were named 'input', 'baddata', 'calculation', and 'output'. The following subsections of the report will focus on the work done on the 'calculation' method only.

3.1.1 Springs

The 'springs' section includes three buttons which assist with calculations for 1- and 2-stage coil springs. However, the inclusion of torsion bars and progressive springs is an attractive possibility for future versions of the calculator.

For 1-stage springs, the calculator assumes that the user already knows the required spring stiffness, which may have been found by simple division or from another process in the calculator. Given the spring stiffness and material parameters, the calculator allows the user to calculate different configurations of coil diameter, wire diameter, and the number of coils that will produce this spring stiffness. Alternatively, the calculator can be used to find the spring stiffness of a coil spring given its geometric and material parameters. This method also allows for calculations of pre-tensioned springs, details of which will be found in the informational window.

For 2-stage springs the calculator has two buttons. One of the two buttons calculates the required spring stiffness of each of the two springs, and their maximum displacements. In order to calculate these values, the user enters the desired displacement (X_1, X_2) when two different forces of their choice are applied (F_1, F_2). The user must also specify either the force or displacement at which they wish the first spring to bottom out. This way, they can increase or decrease the range over which the spring performs linearly.

Using this data, the program produces a graph as seen in Figure 2. By then finding the gradient of the lines, the effective stiffness (k_{eff}) and the stiffness of spring 2 (k_2) can be calculated, and the spring stiffness of spring 1 (k_1) can also be found using these values. With the derived data, the maximum compression of the two springs can be solved quite easily.

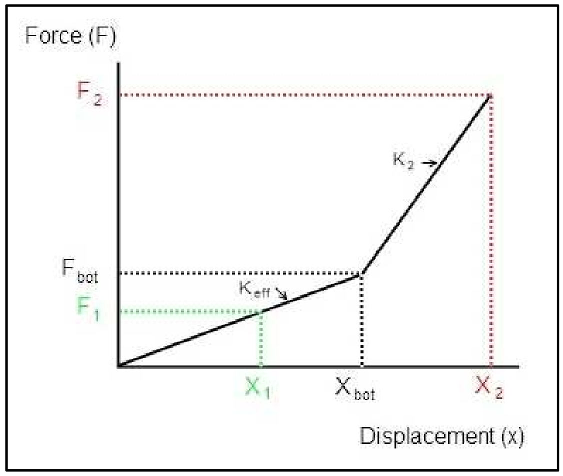


Figure 2: Force vs. displacement graph for a 2-stage spring

The second button relating to the 2-stage springs allows the user to find and specify the desired geometric parameters of the spring which would attain the performance seen in Figure 2. To perform this calculation a similar method to that of the 1-stage spring has been employed. This is too lengthy to outline for the purpose of this report.

3.1.2 Wheel Loads

The 'wheel loads' section is the most comprehensive, using 8 buttons to perform all the calculations. Seven of the buttons allow the user to enter vehicle data to calculate wheel load changes due to different effects. The last button then allows the user to output all the wheel load changes on one screen. It also allows for a few of the parameters such as lateral and longitudinal acceleration to be changed, in order to quickly see the effect that this has on the various wheel load changes. Originally, this last button also interpreted all of these wheel load changes and outputted the final dynamic wheel loads. However, it was decided to remove this feature, as the data can be quite complicated to interpret when there are multiple wheels that are outputting negative loads. Instead, a detailed explanation of how to interpret the data will be provided. Then using the overall wheel loads that are found for each wheel, the forces in and on individual components throughout the suspension system can be calculated.

The seven buttons that calculate load transfers due to various effects do so for the following effects; lateral load transfer, longitudinal load transfer, kerb strike, banking, grade and aerodynamic. Each of these effects has its own button, with the exception of lateral load transfer, which has two buttons. One button calculates the lateral load transfer using a 2-mass model and the other using a 1-mass model. Originally, it had been planned to only include the 2-mass model. However, as this model requires the CoG height of the unsprung mass, which is often unknown, a single mass model, which does not require this value, was also used.

The theory involved in the calculations for this section, varies from button to button, but in many cases is related to or predominantly based on taking moments about an axis and solving. To illustrate this, the longitudinal weight transfer will be found by taking moments about O in Figure 3. Taking moments yields:

eq. (1)

– eq. (2)

Thus, the change in load of the front wheels (ΔW_x) is found using eq. (2).

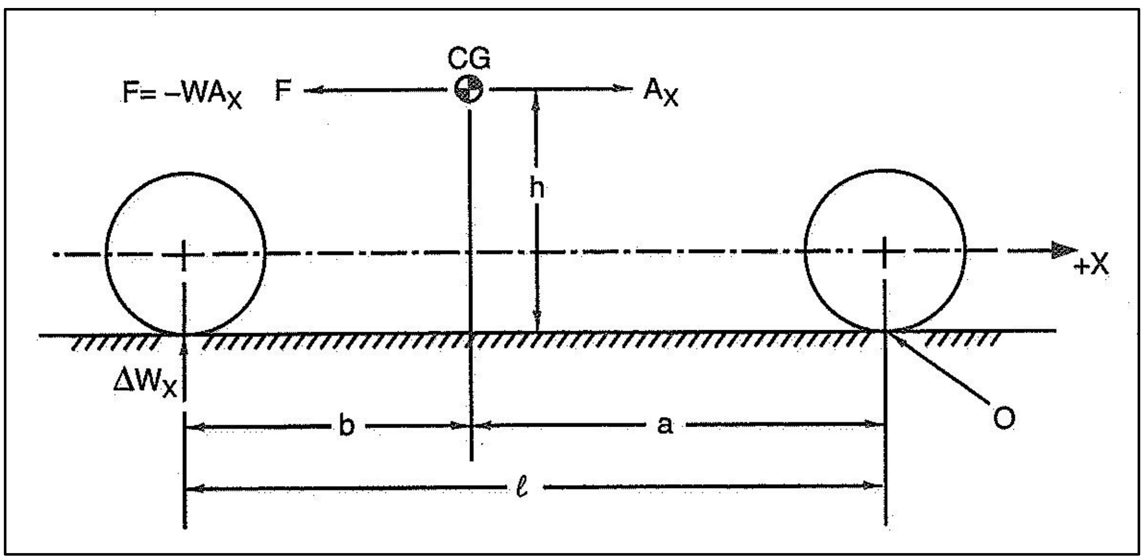


Figure 3: Longitudinal weight transfer (15)

3.1.3 Roll rates

There is only one button that relates to the 'roll rates' section, which is the 'roll and ride rates' button. The user has to input 15 vehicle parameters and then the software uses iteration alongside a few formulae to find the outputted values.

Iteration is needed because the roll rate values are interdependent with the load transfer variables. Thus, the program first estimates the roll rate values, then using these values, calculates the wheel load transfers. The wheel load transfers are then used to calculate the roll rates, and this process is repeated until the variation between iterations is smaller than one thousandths of a percent. Once the iteration is complete, the calculator uses the values found to also calculate the ride frequencies, roll gradients and wheel centre rates.

3.1.4 Wishbones

This section consists of two buttons; ‘calculate forces on wishbones’ and ‘calculate forces on spring and rocker’.

For the ‘calculate forces on the wishbones’ button, the forces at the 6 points in Figure 4 are calculated, as well as the forces at both ends of the pushrod. These are found from the geometry of the system and the tyre reaction forces. The analysis is carried out by assuming that the forces and moments found at points 3 and 6 in Figure 4 would counteract the tyre reaction forces and that the pushrod would take the full vertical load, then resolving the dynamic forces into three force components at each point and solving by equating these in the x-, y-, and z- planes respectively.

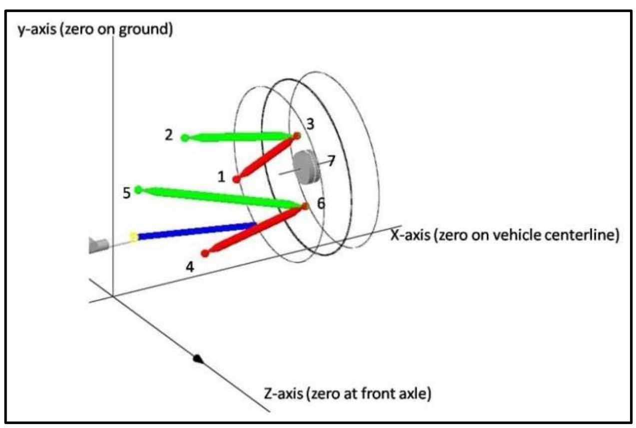


Figure 4: Front left suspension geometry with hard point numbering

The wishbone calculations proved to be far more complicated than originally thought, and the analysis carried out still has room for improvement. One problem with the analysis is that it does not take dive effects, which change certain loads, into account. It also has the problem of being only as accurate as the tyre reaction forces. Finally, the assumption that the pushrod takes the full vertical wheel load becomes inaccurate if the wishbones are not attached horizontally. Given these factors, it is aimed to find a way to improve this analysis.

The ‘calculate forces on spring and rocker’ button uses vectors to find the perpendicular distance of the pushrod and spring axis from the rocker drum axis. The distances are then used together with the pushrod-to-ground inclination to find the motion ratio of the system. Furthermore, this data, in combination with the wheel loads, allows for the spring force to be calculated. Finally, the rocker force is calculated, by assuming that the sum of its vector and that of the spring and pushrod force equals zero.

3.1.5 Imperial Units

The calculator has been created using SI units, and this section aimed at adapting it so that the user could also perform all the calculations in Imperial units. Although, this would be an elegant feature, the intention was to only include it if time permitted. This was because it is a very time-consuming task and not amongst the most crucial to the project. As the project was delayed by two weeks on the date for which work on this section was scheduled, it has been decided to exclude it for the time being.

3.1.6 Unforeseen

There have been unforeseen difficulties and additions within most of the sections of the project. One unforeseen element has been the creation of three completely new sections within the calculations sections, which are discussed below.

3.1.6.1 Anti-roll bar

This section includes two buttons.

The first button calculates the required anti-roll bar roll rate, when given the desired overall car roll rate and the current wheel centre rate. This analysis requires a wheel centre rate that can be found using the roll and ride rate buttons and assumes a linear spring with a constant motion ratio.

The second button calculates which configuration of lever arm length, track width, installation ratio and anti-roll bar angular rate will achieve the required anti-roll bar roll rate. However, the equation can be rearranged so that any of the 5 variables can be found when the other 4 are specified.

3.1.6.2 Clear data

An additional button to clear all data was also included. This button allows the user to reset all values at the click of a button, useful for if they wish to calculate values for a new configuration.

3.1.6.3 Centre of gravity

The CoG is needed to perform many calculations in the program and for this reason three buttons for calculating it were included, despite initially not having been planned for. The first button finds the horizontal location of the CoG, the second button calculates the vertical location, and the third finds the location of the sprung CoG. The user simply enters a few accessible geometric and mass measurements, and in the case of the third button, also enters the horizontal parameters found using the first button. The calculated values in this section are found by resolving moments about an axis, as illustrated in section 3.1.2.

3.2 User Interface Design

This section is currently behind schedule due to the delay in the calculations section. However, some work has been started, which is detailed below.

3.2.1 Window Layout

The window layout within each of the 19 buttons should be as similar as possible, in order to improve user-friendliness. Therefore, consistency has been aimed for throughout the project. Each window will have five components organized in a similar way as seen in Figure 5; input data, output data, buttons/user controls, informational text and an informational diagram. However, this format has yet to be applied to most windows. Moreover, the window seen in Figure 5 is still not a final version of the layout, as it does not include such content as an informational button and it has not been designed to run on screens of varying resolutions. Thus, the majority of the work in this section is still outstanding.

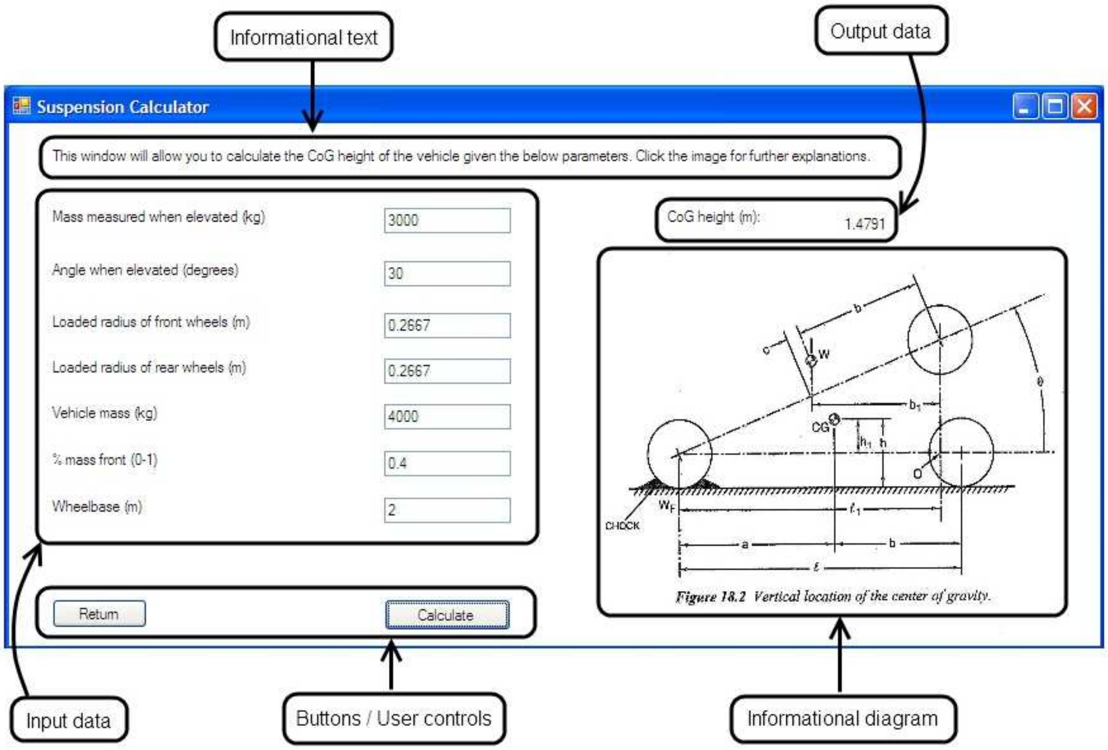


Figure 5: Example layout of windows in program

3.2.2 Informational Windows

Each button will have an informational icon, which if pressed will launch a new window that provides textual and visual information to aid the user. So far research has been carried out on available graphics software, and PaintNET (8) has been chosen for the creation of the images. A few images have been found that will be used in the program when redrawn and modified slightly. However, no informational windows have currently been completed, and focus will shift to this section shortly.

3.2.3 Graphics

The graphics section refers to the graphics of the windows in the program, for example, the use of a background theme or the icon used. Work in this section has not started yet.

3.3 Deployment

Work in this section was originally scheduled to start on February 22, 2010. However, this has been rescheduled as explained in section 4 and as such, some work has been done on the website section instead.

3.3.1 Website

The website can be found on http://www.suspensioncalculator.com/ where a very basic site can be seen with a link to the executable file for the program, and a link to its twitter page. (9) So far the site has received 104 visits from 27 countries, mainly attributable to discussion on the Formula SAE forums. (10)

The work that has been done in this section includes the purchasing of the domain (11) and a hosting service, (12) the decision to use Adobe Dreamweaver CS4, (13) the creation of twitter and Google analytics accounts, (14) and some research into search engine optimization. To complete this section will require learning how to use Adobe Dreamweaver CS4 in order to produce an informative, elegantly presented website, as well as spending some time making other people aware of its existence through word of mouth, online forums, and other methods.

3.3.2 Feedback and testing

This section involves beta testing, which will test the integrity of the program and its calculations more thoroughly than was done in the calculations section. Users will be allowed to test the calculator, with the intention of receiving useful feedback, such as the identification of errors, and suggestions on layout and other possible. This also has the additional benefit of attracting others to the project. Currently, no work has been carried out in this area, but will begin shortly and be ongoing until the end of the project.

4 Timeline

The differences between the planned and executed timeline are shown in Figure 6. The ‘calculations’ milestone was reached around three weeks late, on January 28, 2010, due to unforeseen additions in content and the difficulties encountered, as discussed in section 3. The planned tasks within ‘calculations’ were completed two weeks behind schedule. To compensate for potential delays, a blank timetable over the Christmas break had intentionally been left, and the first of these unallocated weeks was used to catch up. However, a further unallocated week and the first of the second term were also used for the unforeseen tasks described in section 3.1.6. One week was then spent improving the ‘wheel loads’. The final week was spent testing data and correcting miscellaneous mistakes. Due to the delay, it was decided to leave out the work on ‘imperial units’ for the time being, as it is a time-consuming and non-vital section. Another result of the delays has been to postpone work on the ‘user interface design’, on which one week has been spent. Work on ‘deployment’ has also started ahead of schedule for practical reasons.

Figure 7 presents a revised timeline that takes the delays into account and shows some reordering of tasks. The website has been prioritized as all the current work will not be put to its intended use if it is not deployed. Furthermore, in this way feedback is no longer restricted to two weeks at the end of the project, but extends from week 20 to 25, which will allow for the creation of a better product. Consequently, the user interface design section’s deadline has been delayed by 3 weeks, and has been allocated 4 weeks for completion in contrast to the original 6 weeks.

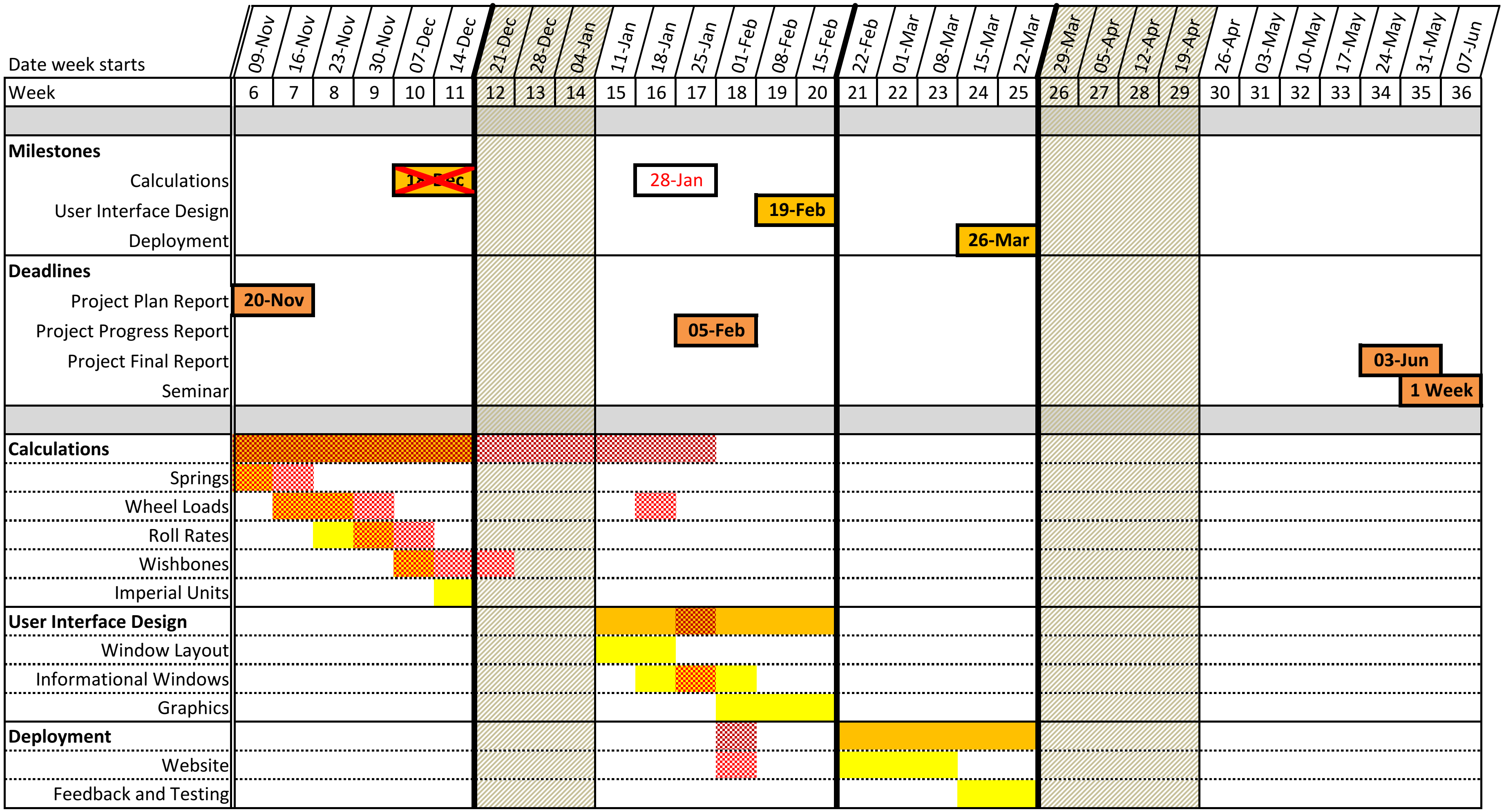


FIgure 6: Plan and progress comparison

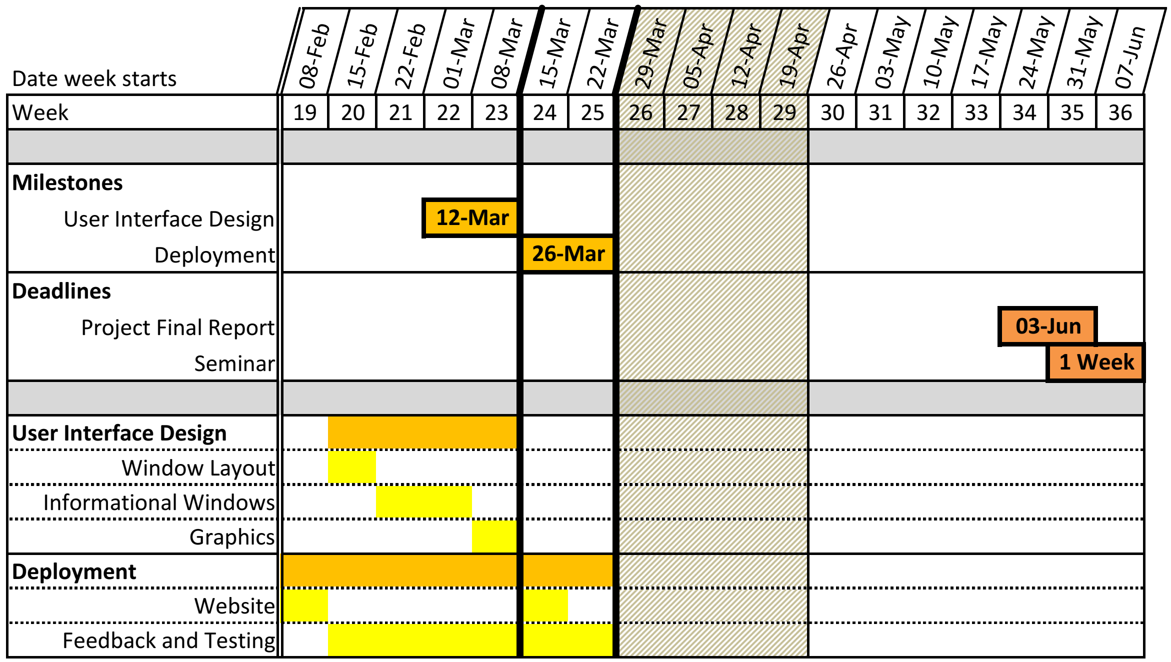


FIgure 7: Revised Project Plan Timeline

5 Conclusions

Over the course of this project several unanticipated difficulties have been encountered, such as the complications in the wishbone calculations, and the original exclusion of the anti-roll bar calculations in the Gantt chart. These difficulties have created extra work that could not always fit into the already tight schedule, which resulted in an approximate three week delay. It is important to make sure that this delay is compensated for and to leave enough room for error in case some other unexpected difficulties crop up at the end. Although this will not be easy, there is the benefit that it was set out to complete two thirds of the project in the first term and one third in the second. This decrease in work in the second term, alongside the crucial restructuring of the tasks as explained in section 4, should allow for the product to be fully completed on time, or completed with minimal compromises.

The use of the Gantt chart has been beneficial throughout the project. Despite it being difficult to judge the time each task would take, and sometimes requiring a fair amount of flexibility, the project has managed to be kept close to schedule. The Gantt chart's advantages are not limited to setting out predetermined time windows. It has been helpful in structuring the project work, having something to benchmark progress against and clarifying the targets.

6 References

1. **Liddiard, Lowery, Schey, Wright.** *Suspension Group Final Report.* London : s.n., 2009.

2. **Microsoft.** *Visual C# 2008 Express Edition.*

3. **Lowery, Cristobal.** *Project Plan Report.* London : Imperial College London, 2009.

4. **Foxall, James.** *Microsoft Visual C# 2005 in 24 Hours, Complete Starter Kit.* Indiana, USA : Sams Publishing, 2006.

5. **Jones, Bradley L.** *Sams Teach Yourself C# in 21 Days .* s.l. : Sams, 2001. ISBN: 0672320711.

6. **FunctionX, Inc.** C# 3.0. *FunctionX Tutorials.* [Online] FunctionX, Inc., 2009. [Cited: 20 10 2009.] http://functionx.com/.

7. **QuackWare.** C Sharp (C#) Tutorials by QuackWare. *Quack-Ware.* [Online] [Cited: 20 10 2009.] http://www.quack-ware.com/tutorials/CSharp.aspx?page=1.

8. **dotPDN.** *PaintNET.*

9. **Twitter, Inc.** *Twitter.* [Online] http://www.twitter.com/.

10. **FSAE.** FSAE forums. *FSAE.com.* [Online] FSAE. [Cited: 20 10 2009.] http://fsae.com/groupee?s=763607348&cdra=Y.

11. **Webfusion Ltd.** *123-reg | Domain names | Domain name registration.* [Online] http://www.123-reg.co.uk/.

12. **Yetihost.** *Yetihost - Provider of quality Coldfusion MX 7 Hosting.* [Online] http://www.yetihost.com/.

13. **Adobe.** *Dreamweaver CS4.*

14. **Google Inc.** *Google Analytics | Official Website.* [Online] http://www.google.com/analytics/.

15. **Milliken, William F. Milliken and Douglas L.** *Race Car Vehicle Dynamics.* Warrendale : Society of Automotive Engineering, Inc., 1995. ISBN: 1560915269.

Throughout this project 44 unique sources have been used; however, this exhaustive list has been left out of this report.