DESIGN CHALLENGE Virtual App (Beta) User Manual 2021

IMechE DESIGN CHALLENGE

REPEATABLE DEVICE

Key HQ Contact: Jelena Gacesa, Operations Manager – Education Programme

jelena.gacesa@imeche.org

designchallenge@imeche.org

Tel: 020 7304 6867

Please check the IMechE Design Challenge website for updates.

Contents

List of Amendments:

Version	Page	Details	Date
1.0	-	Full Release	02-03-2021

1. Introduction

The IMechE Design Challenge app has been developed by the Institution's volunteers to provide participating teams with the ability to test their repeatable vehicle on a virtual platform. The need arose from Covid-19 preventing many teams from building physical devices. The app uses *Unity*, which is a game-engine software. If you're into gaming, then you have probably played a game that has been developed with *Unity*.

In simple terms, data representing the physical characteristics of your design will be used to drive a virtual representation of your repeatable vehicle. What you will be seeing is a visualisation of F=ma (although it's a bit more complicated than that behind the scenes!). It's a good example of how you can use engineering mathematics to simulate reality. For simplicity, and only in this Beta version, the app uses an arbitrary device for the graphics (shown in the image on the title page). For the Regional Finals you will see your team's device in virtual reality.

2. How to Use the App

This document and the accompanying Quick Start Guide include full instructions of the four steps you will need to follow when using the Design Challenge app:

- Read this manual first.
- Download the Beta app.
- Modify the data file to match your team's device.
- Run the app.
- Look at the results file.

3. What You Will Need for the Beta Version

The Design Challenge app has been designed to model the most common type of vehicle. It has not been possible to create an app to simulate every type of repeatable vehicle. To run the IMechE app, your team will need a device that runs on wheels and is powered by an electric motor. The wheels can be driven directly by the motor, or through a gear ratio.

For the Design Challenge app, you will need to know: the mass of your vehicle, the wheel size and mass, the motor characteristics, overall gear ratio, and the frontal area of your device. A detailed explanation of the required information is provided later. You can take these values from your CAD design, but if you are not at that stage yet, you could use the app to try to vary the parameters, to see what happens.

4. Regional Final Requirements

For competing in a virtual Regional Final, your team will need to submit a full CAD model of your design. This is needed for scrutineering and so that your model file can be loaded into the Final app for the competition. Details and requirements for this will be released separately. You will be required to submit your files two weeks before your Regional Final.

Note for organisers of Regional Finals: scrutineering and running the Final app will be handled by the IMechE. Please liaise with Jelena Gacesa and she will help organise the logistics for your final and running of the event on the day.

5. What's in the App?

Here is a comparison of the functionality between the Beta version and Final version of the IMechE Design Challenge app. The Beta version is available now for your own development. The Final version will be used for teams entering the virtual competition at regional level, but it will not be released for general use, to prevent teams gaining a competitive advantage.

Description	Beta Version	Final Version	Notes
Graphics of your device	✖	✔	The final version will have a visual representation of your device
Full CAD model of your device	✖	✔	Estimated data can be used for Beta testing, but a CAD model is required for the final
Bill of Materials	✖	✔	You will need to submit the Bill of Materials (see IMechE Rules)
Variable range distance	✖	✔	Fixed 1.5m wall distance for Beta version but variable for the final (see IMechE Rules)
Variable lane angle	✖	✔	Set at 0° for Beta version but variable for the final (see IMechE Rules)
Motor characteristics	✔	✔	The simulation relies on values for stall torque and no-load speed
Motor efficiency	✔	✔	The app applies typical values – the same for all teams
Gear ratios	✔	✔	You can input the overall gear ratio for your device
Wheel inertias	✔	✔	Rotational inertias are calculated from data you input
Rolling resistance	✔	✔	The app applies typical rolling resistance based on the mass of your device
Aerodynamic drag	✔	✔	The app applies typical drag coefficients based on the frontal area of your device
Impact attenuator	✔	✔	The app helps you build an attenuator that will work for your device
Load transfer	✔	✔	The app is a physical model, so it creates load transfer due to acceleration
Measurement of energy used	✔	✔	The app calculates the energy used by your device to complete the challenge
Measurement of time taken	✔	✔	The app calculates the time taken by your device to complete the challenge
Mechanical distance measurement	✖	✖	Too complex to include as every device is different – also see next point below
Measurement of device accuracy	✖	✖	Too complex to include as the accuracy depends on real, physical build quality
Alternative drive mechanisms	✖	✖	The app requires an electrically driven device with wheels and a motor

6. How the App Works

Have you ever noticed in some driving games that the car appears to 'drop' onto the track when you restart a lap? This is because of the way a game-engine works. When you run the IMechE Design Challenge app, you will notice that the device pauses momentarily before it starts to move. This is because the device is dropped into the scene and the pause allows the model to stabilise before the motor starts. The device then heads for the wall and calculates its way based on the mass, inertia, motor and gear characteristics, etc. that you supply. These values are input into a data file, which will be covered in more detail in Section 13.

Please note that for simplicity the Beta app has a fixed wall distance and horizontal lane. This also prevents teams from 'reverse-engineering' the best possible device through trial and error of all the permutations of range and lane angle before the final.

7. Terminology
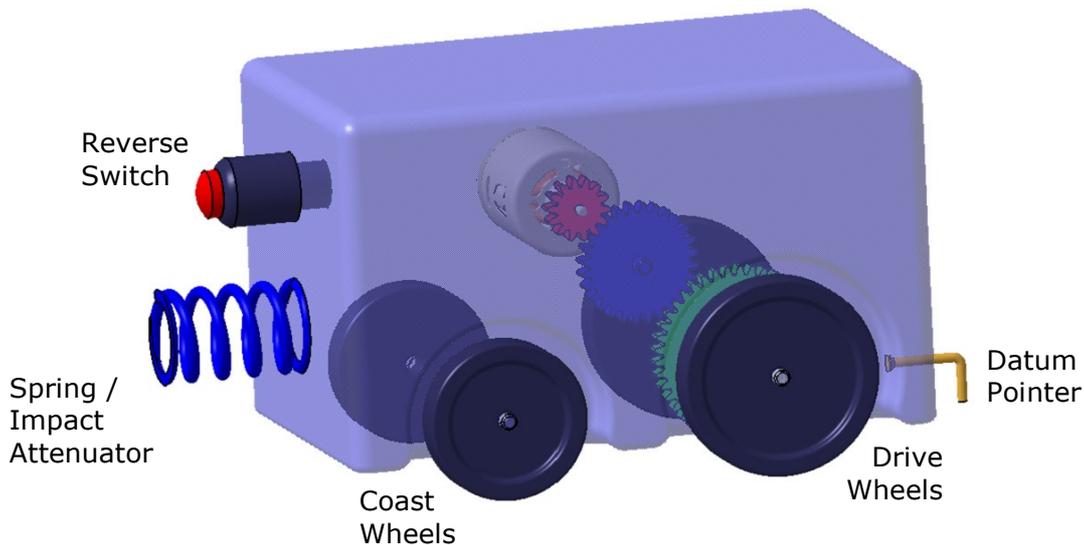
Figure 1: Terminology for Data Input

The positions shown are arbitrary and for reference only. Drive wheels do not have to be at the rear. The spring and switch do not have to be mounted on the front surface of your device. Any dimensions required are relative to the datum pointer, as detailed in Section 13.

The spring and switch are not visible in the Beta app. We recommend you model them for your device, so they appear in your simulation at the Regional Final.

8. Impact attenuator

When the vehicle reaches the wall, it can't be allowed to simply crash into it, because this will make *Unity* unstable. Similar to real life, the behaviour after a sudden impact may be unpredictable. For this reason, you will need to add a simple spring on the front of your device to act as an impact attenuator. The maths is done for you in the IMechE Design Challenge app, so all you need to do is specify the length and stiffness of a single spring. You can do this by trial and error, but some of your own basic calculations may help. Watch out though! The spring will mimic real life. The spring length and constant you choose must be real. For this reason, you will be required to choose a spring that is commercially available and supply the details with your entry for scrutineering (Regional Final). The spring still needs to fit inside the maximum envelope allowed by the Project Specification rules and this will be checked during scrutineering. Details on how to input your impact attenuator data can be found in the Data Input section.

9. Reverse Switch

To add a level of realism to reversing your vehicle, you will need to specify the location of a reverse switch relative to the front of your device and the spring. Handy hint: if your vehicle stops and won't reverse in the app, it may be time to stop and think yourself! Maybe your switch never reaches the wall, if your spring is too stiff, or the switch never reaches the wall, for example? You need the right combination of geometry and spring for the vehicle to work. The switch still needs to fit inside the maximum envelope allowed by the Project Specification rules and this will be checked during scrutineering. Details on how to input your reverse switch data can be found in the Data Input section.

10. Motor

The motor operating characteristics you choose must be real. For this reason, you will be required to choose a motor that is commercially available and supply the details with your entry for scrutineering (Regional Final). If you have already chosen a motor that does not have a data sheet giving the stall torque and no-load speed, you will need to choose another one that does. Details on how to input your motor data can be found in the Data Input section.

These are the only two data points needed for modelling the basic motor characteristics
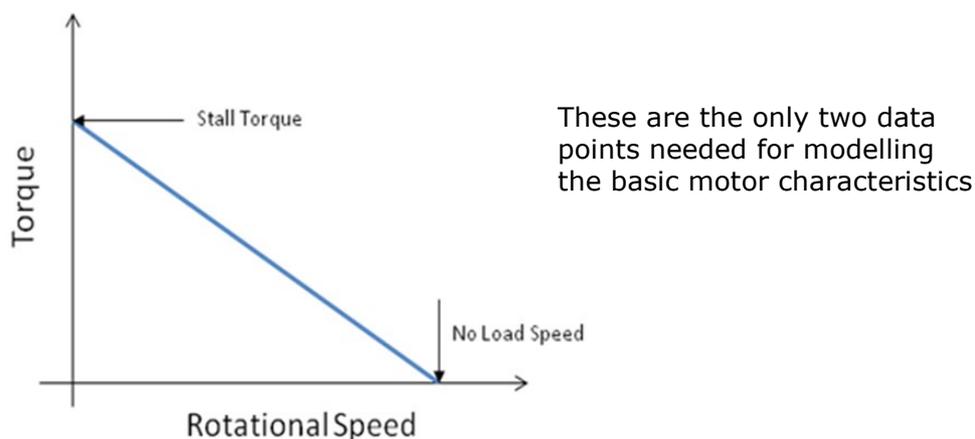
Figure 2: Basic DC Motor Speed-Torque Characteristic

(Image courtesy of https://www.motioncontroltips.com/torque-equation/)

11. Gearing

Depending on the motor that you choose, you may need gears between the motor and the wheels. If the motor is attached directly to the wheels, then the drive ratio would be 1:1, or as an input for the app 1. It is more likely that you will have at least one pair of reduction gears in you drive system. In the diagram below, the red pinion would be attached to the motor and the green gear to the wheels.
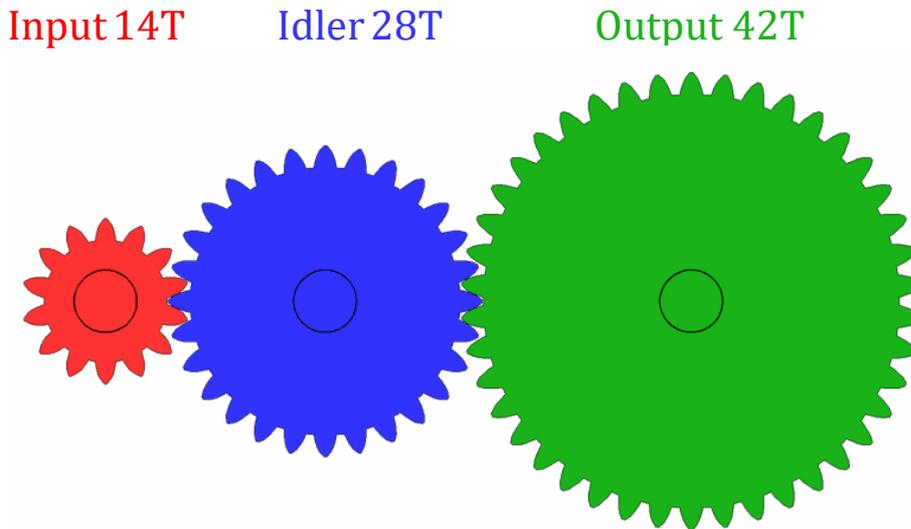

Figure 3: Reduction Gears   *(Image courtesy of https://commons.wikimedia.org/wiki/User:Jahobr/Gearboxes)*

In this example, the reduction ratio is 42/14 so the gear ratio would be input as 3. Whatever your gear set up, another way to calculate this is the ratio of the motor speed to the wheel speed. However, you will need to supply the gear details with your entry for scrutineering (Regional Final). Details on how to input your gear data can be found in the Data Input section.

12. Frontal Area

To calculate the aerodynamic drag you will need to input the frontal area of your device. For the Beta app, this can be an estimate if you choose, but for the Final version, you must provide an accurate value. You can do this by projecting the frontal view of your vehicle and working out the area within the silhouette.
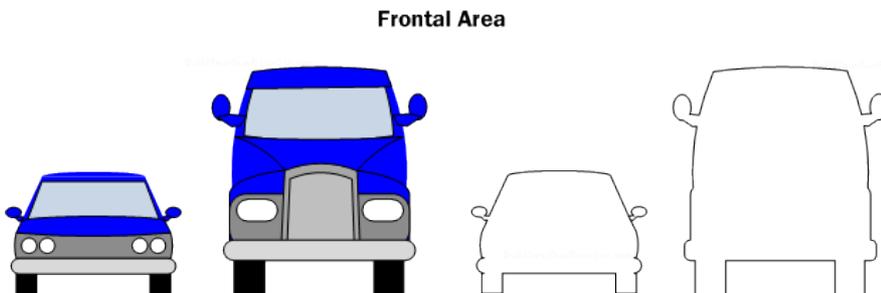

Figure 4: Frontal Area   *(Image courtesy of https://www.buildyourownracecar.com/race-car-aerodynamics-basics-and-design/2/)*
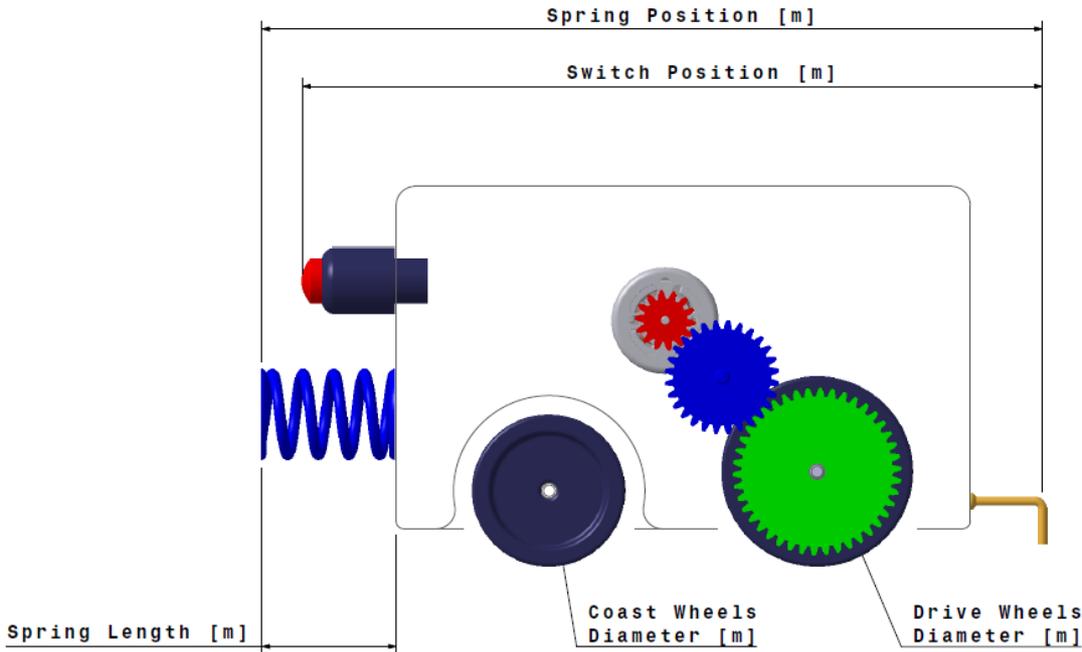
13. Data Input

Figure 5: Dimensions for Data Input

Data for your device is entered into a csv file, as shown below. It is important to edit the csv file with Notepad to conserve the file format for *Unity*. Values can be changed and then the file saved ready to run the app. Please note the units!! If you enter mm instead of m, or g instead of kg, for example, your simulation will be wildly wrong.
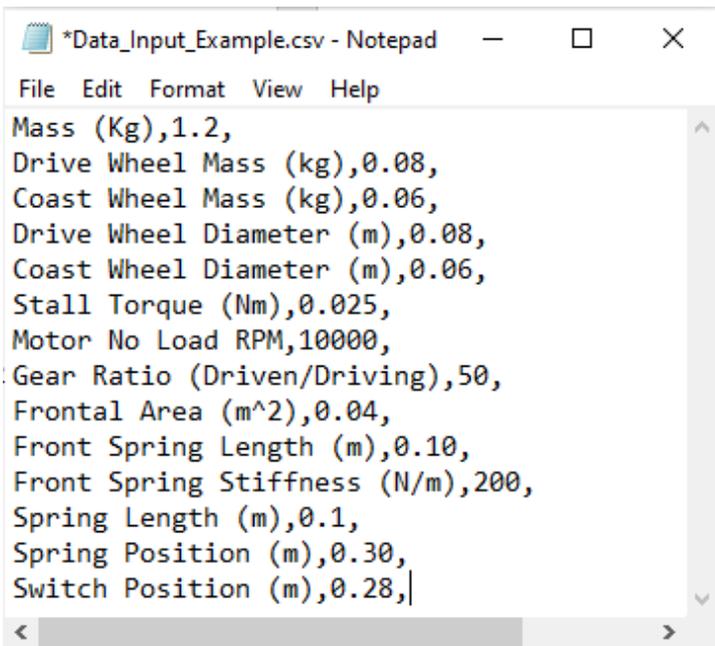

Figure 6: Sample of the VehicleData2 Input File

Here are a few notes on items that have not been mentioned so far:

Mass - this is the mass of your entire device, if you put it on a set of weighing scales. For Beta testing, you can use an estimated mass. For the competition, you will need to work out individual masses, either by applying materials to your CAD model and taking measurements or weighing individual components like motors and batteries. This is exactly what happens in industry – it does take time and effort. See the Project Specification rules for how the individual masses need to be recorded in a Bill of Materials.

Drive Wheel Mass and Diameter – this is the mass of a single wheel. The drive wheels are the ones driven by the motor. The diameter is the rolling diameter of the wheel.

Coast Wheel Mass and Diameter – this is the mass of a single wheel. The coast wheels are the ones that free-wheel. The diameter is the rolling diameter of the wheel.

The app presumes you have two drive wheels and two coast wheels. If you only have a single drive wheel, or single coast wheel, halve that wheel mass so that the app works for your device.

Stall Torque – this is the torque of the motor at zero rpm. You will be able to find this information on the manufacturer's website.

No-load speed – this is the speed of the motor at zero torque (when it is spinning freely without being attached to anything). You will be able to find this information on the manufacturer's website.

If you cannot find values for the stall torque and no-load speed for your motor, it suggests that you are not purchasing from a reputable supplier. In reality, you wouldn't be doing your job properly if you didn't have this information.

The app includes some simplifications to allow it to work. For instance, it does not include all the rotating inertias involved such as the motor and gears. This is because it would be difficult to get teams to enter individual gear and motor inertias and their effect is not significant compared to the wheel inertias (see Managing Expectations).

14. Installing and Running the IMechE App

Please see the accompanying document for full details:

IMechE Design Challenge - Virtual App (Beta) Quick Start Guide 2021 - V2.9

You may find that the Design Challenge app stutters, depending on the spec of machine you run it on. Please check the Quick Start Guide for information on how to avoid this.

15. Results Output and Analysis

Please see the Quick Start guide for full details, but here are a few tips when looking at the results:

Delete the excel file ExportData.csv when you want to process a clean set of data (*Unity* keeps writing to the same file otherwise).

We recommend you set up a template Excel file to process the results with graphs and then copy and paste results into the template. This will save time and effort.

The device starts moving after 2.0 seconds. Make sure you take account of this in your data processing. The delay at the start is to allow the device to stabilise and also for the camera to pan into position for the visualisation.

16. Requirements for the Regional Final

Teams will be required to submit files two weeks before their Regional Final to allow for scrutineering and setting up the Final app for each competing team. You will also need to submit you CAD model. Please see the IMechE Project Specification document for full details.

The IMechE will not be releasing the Final app because it takes a level of expertise to load the CAD models and it also prevents teams from optimising their design through trial and error.

17. Managing Expectations

Please bear in mind that the IMechE Design Challenge app has been created by volunteers working in their spare time. The motivation has been to give challenge teams a better experience. You may have heard of the 80-20 rule (the Pareto Principle)? The majority of the outcomes of a project can be achieved with a relatively small input. This is true for the app because 80% of the functionality is there and it simulates a repeatable vehicle. However, it is not as slick as something you would normally download from an app store. To make it a fully developed app, with a glossy interface, and things like play buttons, a user interface for data input and results processing, would take a significant amount of time and effort. Remember this when you are entering data using Notepad and looking at the results in Excel! Likewise, decisions have had to be made to strike a balance between realism and accommodating many different designs. Simplifications and assumptions have had to be made as a result.

18. FAQs and Help

Help Webinars
The IMechE will announce the times of online webinars where the Beta app will be introduced, and competitors can ask the app developers any questions they have.

Mac Users
Because of time and resource limitations, it has only been possible to develop the app for use with Windows.

What if the app doesn't work?
If the device fails to move, this suggests that one or more of your inputs is unrealistic. Please check you have entered sensible values using the correct units.

What if the device falls off the lane?
This happens when *Unity* becomes unstable. If your device behaves erratically or is unstable this suggests that one or more of your inputs is unrealistic. Please check you have entered sensible values using the correct units.

What if the device stops at the wall?
The device will stop at the wall if the spring and switch characteristics are wrong. This usually happens because the reverse switch has not been triggered. Either your spring is too long, or it is too stiff, or the reverse switch is too far back relative to the spring.

The device doesn't appear to hit the wall?
This is because the spring isn't modelled visually in the Beta version. Mathematically, though, the app is working out how the device will rebound for the length of spring you have given (small technicality - the minimum distance to the wall will not necessarily reach zero).

The device appears to overshoot the target
This is because the motor is switched off at the target distance, but your device skidded, as it would in reality, and has overshot the target. *Unity* is calculating the overshoot but you will notice that the energy use doesn't increase once the motor has disengaged.

How to get More Help
If you continue to have problems, please send emails to *designchallenge@imeche.org* along with your VehicleData2.csv file.