Group 9: Optimal Design of a Rollercoaster

Owen, Sophie — *Sub-system 1: Cart*
Thompson, Oliver — *Sub-system 2: Drop*
Last, Matthew — *Sub-system 3: Loop*
Tiersen, Federico — *Sub-system 4: Zero-G Roll*

ABSTRACT

This paper details an optimisation study on the design of a non-inverted, traditional rollercoaster. The goal of the study is to maximise passenger enjoyment by minimising the objective functions of 4 subsystems. Passenger enjoyment has been defined through research as high speed, high G-forces and unnatural experiences such as prolonged weightlessness. These can be described by various physical parameters of the track and rollercoaster.

The overall system level optimisation used a combination of nonlinear solvers from each subsystem, to return values comparable to industry leading rollercoasters.

The mass of the cart was found to be 4100 kg and the coefficient of rolling resistance 0.267, amounting to a rolling resistance of 920 N. These values were fed into subsystem 2, where the maximum velocity, drop height and G force over the drop were found to be 33 ms^{-1} , 76 m and 6 G respectively.

The third subsystem concerns a loop in the track of the rollercoaster. Its curvature is optimised to increase enjoyment by maximising the amount of G force the passengers are exposed to, again whist keeping in safe industry boundaries. This was achieved, providing a loop with an average G-Force level of 3.

The airtime experienced with the optimal launch track (length 84.4 m, height 22.5 m, launch angle 31.3 °) is 5.3 s, as found by subsystem 4. Methodologies from subsystems 2,3 and 4 may be combined in further studies to design industry-leading track shapes from desired G-force profiles while avoiding sudden G-force changes.

1. Introduction

This project maximises the enjoyment of a rollercoaster, by optimising various parameters which have been directly linked to enjoyment. These include extreme G-Force [1], accelerations and velocities, [2] as well as a variety of track formations.

The problem is tightly constrained by safety restrictions, particularly staying within safe G-Force limits. This means a balance is needed between maximising speed, acceleration and safety [3].

The physics governing a rollercoaster and the effects of g-force on the human body are well documented [4], as are data about current rollercoasters, making pre-made datasets easy to find. A ride on a rollercoaster containing at least one drop, one loop and one zero g-roll will guarantee all enjoyment factors identified in our research.

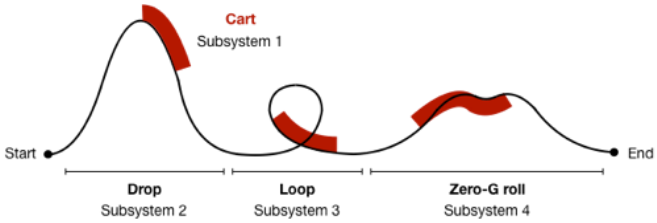
1.1 System-level problem and Subsystem breakdown

$$min \quad f(\boldsymbol{X},\boldsymbol{P}) = [f_1(\mathbf{x_1},\mathbf{p_1}) + f_2(\mathbf{x_2},\mathbf{p_2}) + f_3(\mathbf{x_3},\mathbf{p_3}) + f_4(\mathbf{x_4},\mathbf{p_4})]$$

Where

$\mathbf{x_1} = (,\rho, d_o, d_i, \sigma, E) \in \mathbb{R}$; $\mathbf{p_1} = (W_c, W_p, g, \sigma', E')$
$\mathbf{x_2} = (\dots) \in \mathbb{R}$; $\mathbf{p_2} = (\dots)$
$\mathbf{x_3} = (\dots) \in \mathbb{R}$; $\mathbf{p_3} = (\dots)$
$\mathbf{x_4} = (\boldsymbol{y}, \boldsymbol{\theta}) \in \mathbb{R} \in \mathcal{S}^{Model}$; $\mathbf{p_4} = (\boldsymbol{s}, g, u, G_{max})$

Subject to: the constraints defined in each subsystem optimisation.



Four subsystems are considered as modules of our rollercoaster: the geometries of the drop, loops and zero-g rolls, as well as the features of the cart. The design of the cart has implications on the whole rollercoaster. For example, the number of people per cart affects the impact load and therefore the speed at which the coaster enters each section of track.

The first subsystem analyses the cart. It optimises the efficiency of the wheels by designing them such that the rolling resistance is as small as possible. The dimensions and material of the wheels, and the coefficient of friction are outputs.

The second subsystem uses the rolling resistance returned from the first subsystem to optimise the geometry of the track during the initial drop. The objective function is to maximise velocity as the rollercoaster exits the drop, within the boundaries of safe G force limits and terminal velocity.

The third subsystem uses height returned from the second subsystem to generate a loop that maximises the G-force incident on the passenger, whist keeping within safe G-force boundaries.

The final subsystem looks at the shape of the launch section of a *Zero-G Roll*, an inversion in which passengers experience a feeling of weightlessness. The optimisation maximises the length of this *airtime* while avoiding sudden G-force changes.

As the subsystems have no interdependencies and there are no trade-offs between them, minimising the system level objective function is achieved by minimising each of the subsystems individually; resulting in a rollercoaster with minimal rolling resistance, a maximum drop speed, a loop with maximum G force, and a zero-G roll with maximum air-time.

2. Subsystem 1: The Cart

Rollercoasters are said to be more enjoyable when the cart can reach a high maximum velocity, as this makes accelerations faster and they experience more fluctuations in G-force [3]. The higher the speed for a given track length, the quicker the ride and the shorter the queue time. Therefore, the cart must be designed for the highest possible speed, by increasing the efficiency of the cart components.

Subsystem 1 focussed on optimising the wheels. The aim is to reduce the rolling resistance of the wheels so minimal energy is wasted to overcome friction, resulting in a higher maximum possible velocity.

2.1 Optimisation formulation: Summary

$$min \quad f(w,\rho,d_o,d_i,\sigma,E) = \sqrt{\frac{\left(\frac{3}{2\pi w}(W_c+W_p)+\frac{3\rho g}{8}(d_o^2-d_i^2)\right)\left(\frac{1-\sigma^2}{E}+\frac{1-\sigma'^2}{E'}\right)}{d_o}} \cdot \left(W_c+W_p+\frac{\pi\rho w g}{4}(d_o^2-d_i^2)\right)$$

where $\mathbf{x} = (w,\rho,d_o,d_i,\sigma,E)$

$$\mathbf{p} = (W_c, W_p, g, \sigma', E')$$

s.t.
$$h_1(W_c)\text{: } W_c = 2624.175$$
$$h_2(W_p)\text{: } W_p = 981$$
$$h_3(\sigma')\text{: } \sigma' = 0.295$$
$$h_4(E')\text{: } E' = 200 \times 10^6$$

$$g_1(w)\text{: } w - 0.08 \leq 0$$
$$g_2(w)\text{: } 0.01 - w \leq 0$$
$$g_3(d_o)\text{: } d_o - 0.1775 \leq 0$$
$$g_4(d_o)\text{: } 0.03 - d_o \leq 0$$
$$g_5(d_i)\text{: } d_i - 0.06 \leq 0$$
$$g_6(d_i)\text{: } 0.01 - d_i \leq 0$$
$$g_7(d_o, d_i)\text{: } d_i - 0.25d_o \leq 0$$
$$g_8(d_o, d_i)\text{: } 0.1d_o - d_i \leq 0$$
$$g_9(w, d_o, d_i)\text{: } \frac{\rho w \pi}{4}(d_o^2 - d_i^2) - 1 \leq 0$$
$$g_{10}(w, d_o, d_i)\text{: } 0 - \frac{\rho w \pi}{4}(d_o^2 - d_i^2) \leq 0$$
$$g_{11}(w, d_o, d_i)\text{: } w - 0.6d_o \leq 0$$

2.2 Modelling approach

The mathematical model is derived from first principles.

2.2.1 Exploring the Problem Space: Derivation

Rolling resistance is the force required to overcome friction. It is specific to wheels rotating as they have a very small surface area in contact with the ground at any one moment. The equation for rolling resistance [5] is a function of the coefficient of rolling friction (C_{rr}) and the load on the wheel (W).

$$F = C_{rr}W$$
$$C_{rr} = \sqrt{\frac{z}{d_o}}$$
$$z = \frac{2WD}{\pi w}$$

Where D is a function of the Poisson ratio of the wheel (σ) and the track (σ'), and Young's Modulus of the wheel (E) and track (E').

$$D = \frac{3}{4}\left(\frac{1-\sigma^2}{E} + \frac{1-\sigma'^2}{E'}\right)$$

Load on the wheel is the sum of the weight of the cart (W_c), passengers (W_p) and wheel (W_w).

$$W = W_c + W_p + W_w$$

The wheel is modelled as a solid cylinder with a cut out. This is a simple design, much like other wheels available on the market [6].

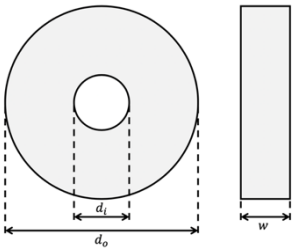


Figure 2.2.1.3 Diagram of the Wheel Dimensions

The Load on the wheel can be broken down further, by calculating the load from the wheel itself.

$$W_w = \frac{\pi \rho w g}{4}(d_o^2 - d_i^2)$$

The equation for rolling resistance can then be reformulated using equations for the weights and resistance coefficient:

$$\sqrt{\frac{\left(\frac{3}{2\pi w}\left((W_c + W_p) + \frac{3\rho w g}{8}(d_o^2 - d_i^2)\right)\right)\left(\frac{1-\sigma^2}{E} + \frac{1-\sigma'^2}{E'}\right)}{d_o}} \cdot \left(W_c + W_p + \frac{\pi \rho w g}{4}(d_o^2 - d_i^2)\right)$$

2.2.2 Material Selection

Young's Modulus was plotted against the Poisson ratio for all materials (**Error! Reference source not found.**). Using CES, constraints were added to filter down the options based on the performance values of polyurethane; the current wheel material for rollercoasters (Table 2.2.2.1). The selected material had to have the same or higher performance requirements than PU.

Table 2.2.2.1 Material selection constraints

$1e10^3 \leq \rho \leq 2e10^3$	Density (kg/m^3)
$38 \leq \sigma_y$	Yield Strength (MPa)
$48 \leq \sigma_c$	Compressive Strength (MPa)
$14 \leq \sigma_f$	Fatigue Strength at 10^7 Cycles (MPa)
Acceptable, Excellent	Fresh Water Durability
$CO_2 \leq 8$	CO_2 Footprint, Primary Production (kg/kg)

9 polymers passed these constraints (
Table 2.2.2.2) alongside the property values which determine the objective function. **Error! Reference source not found.** demonstrates the selection process.

Table 2.2.2.2 Mechanical properties of wheel materials **[7]**

Material	**Poisson's Ratio**	**Young's Modulus (GPa)**	**Density (kg/m^3)**
Polylactide (PLA)	0.4	3.58	1.27e3
Polyurethane (PUR)	0.42	2.07	1.24e3
Polystyrene (PS)	0.4	3.28	1.04e3
Polyoxymethylene (POM)	0.407	3.2	1.43e3
Polymethyl methacrylate (PMMA)	0.4	3.23	1.2e3
Polyethylene terephthalate (PET)	0.4	3.01	1.39e3
Polycarbonate (PC)	0.41	2.52	1.21e3
Polyamides (Nylons, PA)	0.42	2.04	1.15e3
(ABS)	0.41	2.75	1.06e3
Average	**0.407**	**2.853**	**1.211e3**

2.2.3 Constraints

Every variable has at least an upper and lower bound. Some of these are based on anthropometric data and others are taken from restrictions on the wheels available [6]. Other assumptions in Constraints regarding Poisson's Ratio and Young's Modulus have been removed as the

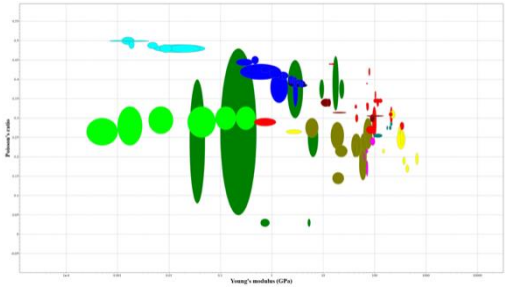


Figure 2.2.1.1 Poisson Ratio vs. Young's Modulus

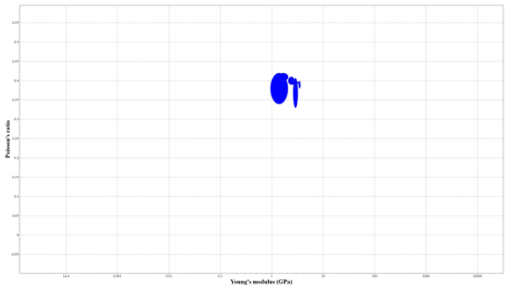


Figure 2.2.1.2 Young's Modulus vs. Poisson Ratio for selected materials

material will be selected from a pre-determined table of materials which have already undergone an initial optimisation filtering process.

Table 2.2.3.1 relate the variables to one other, considering factors such as the inner diameter of the wheel relative to the outer diameter. Constraints regarding Poisson's Ratio and Young's Modulus have been removed as the material will be selected from a pre-determined table of materials which have already undergone an initial optimisation filtering process.

Table 2.2.3.1 Subsystem 1 constraints

g_1	The width of the wheels must be less than 10cm (the width of the tracks).	[8]

g_2	The wheels must be over 1cm thick (the minimum thickness available to buy).	[6]
g_3	The outer wheel diameter must be under 0.3775m, half the average length between a person's knee and the floor when sat, plus the clearance from the cart to the ground.	[9]
g_4	The outer wheel diameter must be over 3cm. This is the minimum diameter of available wheels.	[6]
g_5	The inner wheel diameter must be under 6cm. This is the maximum diameter of available hubs.	[6]
g_6	The inner wheel diameter must be over 1cm. This is the minimum diameter of available hubs.	[6]
g_7	The inner diameter must be less than 25% the outer diameter of the wheel. This is the maximum safe ratio recommended	[6]
g_8	The inner diameter of the wheel must be over 10% the outer diameter of the wheel. This is the minimum ratio possible.	[6]
g_9	The mass must be under 1kg per wheel.	
g_{10}	The mass must be over zero.	
g_{11}	The width of the wheel must be less than half the outer diameter of the wheel.	

2.2.4 Assumptions

Table 2.2.4.1 and Table 2.2.4.2 summarise assumptions made and the logic behind them.

Table 2.2.4.1 Subsystem 1 formulated assumptions

h_1	The weight of the cart, W_c, is taken to be 267.5*g per person. For four people in a cart with four wheels, this accounts to a load of 267.5*g per wheel.	[10]
h_2	The load from one person, W_p, is the maximum load permitted to ride a roller coaster, 100*g.	[11]
h_3	The Poisson Ratio of the steel rails is 0.295.	[12]
h_4	The Young's Modulus of the steel rails is 200 x 10^6.	[12]

Table 2.2.4.2 Non-mathematical assumptions

The wheel is imperfectly elastic; therefore, b is negligible.	[13]
The wheel is simplified and made of one solid material.	
$b \ll w$, the length of contact area between the wheel and rail is much smaller than the width of the wheel.	[13]
$b \ll d_o$, the length of contact area between the wheel and rail is much smaller than the outer diameter of the wheel.	[13]
There are four people per cart.	
There are four wheels per cart.	
The load is split equally between each wheel.	
There are 6 carts, with a total of 24 people per ride.	
The rails are made of carbon steel.	[14]

2.2.5 Linear Analysis

Linear analysis was done to find out if the function was linear or nonlinear, while also giving a better understanding of the problem space. Two values at a time were plotted against the objective function.

It can be seen from Figure 2.2.5.1 that the objective function is non-linear. The width, inner diameter and outer diameter of the wheel all change in a non-linear fashion relative to one another and so nonlinear methods must be used to solve the optimisation problem.

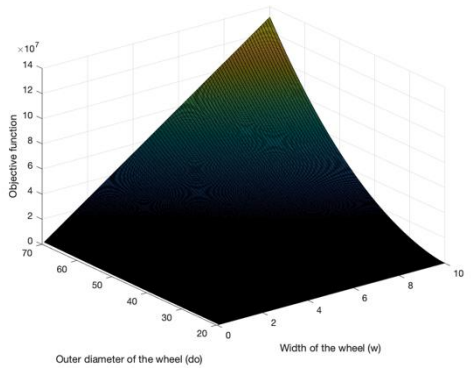


Figure 2.2.5.1 Determining linearity of the objective function.

2.2.6 Monotonicity Analysis

As seen in Table 2.2.6.1, the objective function increases with respect to both w and d_o, but decreases with respect to d_i. It is unknown whether it is increasing or decreasing with respect to ρ, σ, and E. g_{10} is active with respect to w, and g_7 is active with respect to d_o. All variables are constrained from above and below. Inactive constraints are highlighted in grey.

After doing monotonicity analysis it can be confirmed that the problem is well constrained, and the model cannot be simplified further.

Table 2.2.6.1 Monotonicity analysis

	w	d_o	d_i	ρ	σ	E
f	+	+	-	U	U	U
g_1	+					
g_2	-					
g_3		+				
g_4		-				
g_5			+			
g_6			-			
g_7		-	+			
g_8		+	-			
g_9	+	+	-	+		
g_{10}	-	U	U	-		
g_{11}	+	-				

2.2.7 Reformulation of the problem

Following monotonicity analysis, the problem can be simplified.

$$min \quad f(w,\rho,d_o,d_i,\sigma,E) = \sqrt{\frac{\left(\frac{3}{2\pi w}(W_c+W_p)+\frac{3\rho g}{8}(d_o^2-d_i^2)\right)\left(\frac{1-\sigma^2}{E}+\frac{1-\sigma'^2}{E'}\right)}{d_o}} \cdot \left(W_c+W_p+\frac{\pi\rho w g}{4}(d_o^2-d_i^2)\right)$$

where $\mathbf{x} = (w,\rho,d_o,d_i,\sigma,E)$
$\mathbf{p} = (W_c,W_p,g,\sigma',E')$

s.t. $h_1(W_c)$: $W_c = 2624.175$
$h_2(W_p)$: $W_p = 981$
$h_3(\sigma')$: $\sigma' = 0.295$
$h_4(E')$: $E' = 200 \times 10^6$

$g_1(w)$: $w - 0.08 \le 0$
$g_2(w)$: $0.01 - w \le 0$
$g_7(d_o,d_i)$: $d_i - 0.25d_o \le 0$
$g_9(w,d_o,d_i)$: $\frac{\rho w\pi}{4}(d_o^2-d_i^2) - 1 \le 0$
$g_{10}(w,d_o,d_i)$: $0 - \frac{\rho w\pi}{4}(d_o^2-d_i^2) \le 0$
$g_{11}(w,d_o,d_i)$: $w - 0.6d_o \le 0$

2.3 Optimise

As the density, Young's Modulus and Poisson Ratio are independent of the physical dimensions of the wheels, they are initially set to constants. These values are the averages of the material options available. The `fmincon` algorithm was used to solve the optimisation, using both SQP and interior points. The objective function lends itself well to gradient based methods as it is a constrained, nonlinear function derived from first principles.

This was used to optimise the wheel dimensions, w, d_o, and d_i. MATLAB was then used to identify the optimal material for the wheels by calculating the smallest value of rolling resistance. Parametric analysis was then conducted.

2.3.1 `fmincon` Interior Points Algorithm

`fmincon` interior points was used to find the optimal values initially. This found the local minima of the objective function. This method is simple to implement and was fast at solving the optimisation problem. Table 2.3.1.1 shows the results of this method.

Table 2.3.1.1 Results of `fmincon` Interior Points

t	d_o(m)	d_i(m)	w(m)	F_{rr} (N)
0.200743	0.12	0.03	0.05309	1024

`fmincon` can often become stuck at a local minimum, and so to get around this problem multiple initial guesses were tested, and the optimal solution was identified from these.

`Globalsearch` was also used to confirm the identified solution was a global minimum. `fmincon sqp` was also used to check the values.

2.3.2 *`Fmincon` Sequential Quadratic Programming (`sqp`) Algorithm*

To test the reproducibility of the results identified from `fmincon` interior points, `fmincon sqp` was used. This confirmed the correct results. This algorithm ran 6 times and took 0.181446 seconds, slightly faster than `fmincon` interior points. This resulted in a rolling resistance of 1024 N.

2.3.3 *Material Selection*

The materials were optimised using the selected materials from CES. The table of properties, including the Young's Modulus, density and Poisson Ratio, was imported to MATLAB. The objective function was calculated for each material based on these properties and the optimised dimensions found from `fmincon`. Each value was plotted in a graph which can be seen in Figure 2.3.3.1. The minimum value was identified using the `min(obj)` function. PLA was identified as the material with the lowest rolling resistance, so this will be the material for the wheels. This resulted in a final rolling resistance of 919 N.

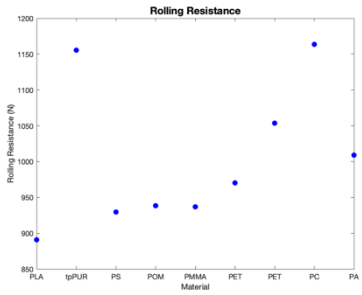
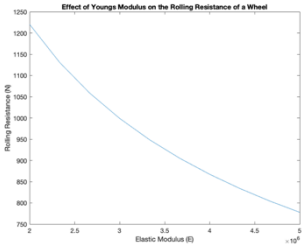


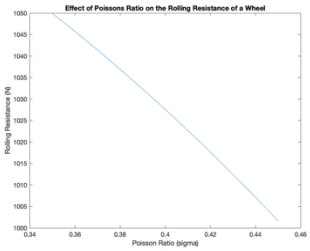
Figure 2.3.3.1 Objective function value for each material option

2.3.4 *Parametric Analysis*

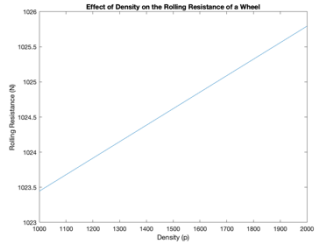
A parametric analysis was undertaken following optimisation by keeping all values constant at their averages and independently changing one or two variables. It can be seen from this that the Elastic modulus and Poisson's ratio have the most effect on the objective function, and the effect of density is nearly negligible in comparison. A surface was plotted from wheel dimension variables to determine their effect the objective function. By looking at the gradient of the surfaces, it can be concluded that both the width of the wheel and the inner diameter have a much smaller effect than the outer diameter of the wheel. When put into the context of a physical wheel this makes sense, as the outer diameter has much more effect on the velocity of the wheel. The inner diameter and width affect the mass of the wheel, which is less significant than the diameter.



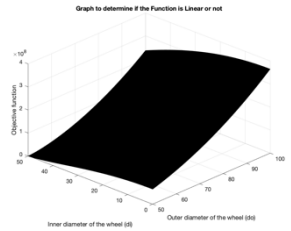
(a)



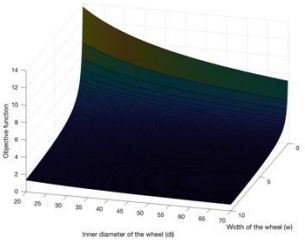
(b)



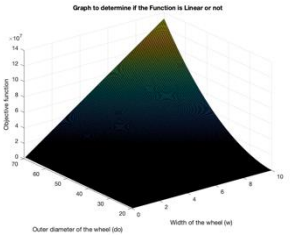
(c)



(d)



(e)



(f)

2.3.5 *Minimising drag*

It is assumed the energy powering the cart is constant. Therefore, to maximise potential velocity, the cart components must be designed for efficiency. To do this, the cart body can be designed to minimise drag.

$$D = \frac{1}{2} C \rho v^2 A$$

C (coefficient of drag) is dependent on the shape of the front of the cart so this can be automatically optimised. Velocity is a constant as the aim is to reduce the drag independently of velocity. The material of the cart is assumed to be steel, so density is also pre-determined. The only variable to be optimised is the area of the front of the cart. This is reduced my minimising the number of people per row, as the height will be designed to accommodate the majority of passengers. However, as the aim of the system is to optimise for enjoyment, and it may be assumed that to ride with a friend makes the experience more enjoyable, there will need to be at least two people per row. Therefore, to maximise enjoyment and minimise frontal area, there will be two people per row, and two rows per cart. The maximum number of carts is 6. To increase enjoyment, the queue time must be minimised, so each ride must accommodate as many people as possible to increase flow. Therefore, the number of carts will be 6 and the ride will accommodate 24 people at a time. This all contributes to the mass of the cart, and the output values to be taken to the system level optimisation are shown in Table 2.3.5.1.

Table 2.3.5.1 Output values from optimisation solution

Variable	Value
Number of people per ride	24
Mass per cart	1.4279e3 kg
Combined mass of all carts	8.8372e3 kg
Load on tracks from the whole train	8.6693e4 N

2.4 Discussion

`fmincon sqp` was the most computationally efficient algorithm, providing optimal results in the shortest amount of time. Following wheel-dimension optimisation, the selection of a new material reduced the rolling resistance per wheel from 1190 N to 919 N.

It was challenging to optimise the wheel dimensions and material selection simultaneously, so the algorithm could be further improved by combining the variables into one function. However, the material and dimensions of the wheels are independent of one another, so they were treated as two separate optimisation problems and the final optimised values are the same regardless.

The optimisation does not take into consideration some factors which would make it applicable in the real-world. To do this, variables such as forces on the side wheels (which keep the cart on the track when it banks) must be considered, alongside forces on the wheels below the track (which stop the cart falling off when upside down).

To develop the optimisation problem further the number of wheels per cart could be added as a variable. This would add an interesting factor to the optimisation as having more minimises the load per wheel, but the increased number of contact points with the track increases combined friction.

3. Subsystem 2: The Drop

3.1.1 Introduction

The initial drop inputs energy into the rollercoaster system. For our purposes the rollercoaster is pulled up a ramp by a chain driven by an engine, and once it crosses the crest of the drop the rollercoaster will accelerate down the slope under gravity. It is important enough energy is given to the rollercoaster to ensure it can complete the rest of the track. To maximise passengers' enjoyment, it is important to maximise velocity. Faster rollercoasters are more enjoyable, and a larger velocity results in larger acceleration (or G force) when the rollercoaster changes direction. The velocity is limited by the terminal velocity of the rollercoaster combined with the bearing friction, and acceleration must be kept within safe limits (both in terms of magnitude and duration) so as not to cause injury. This subsystem will optimise the geometry of the curve after the peak, the drop itself and the curve following the drop (Figure 1).

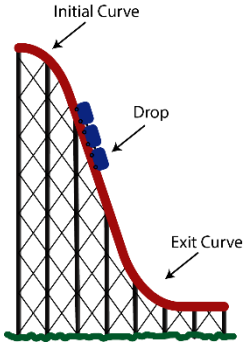


Figure 4.1.1.1

3.1.2 Derivation

The subsystem can be modelled from first principles by combining Newton's second law (equation 1), centripetal force (equation 2), force due to friction (equation 3) and force due to air resistance (equation 4)

1: $m \cdot a = g \cdot \sin(\theta) - F_{air} - F_r$

2: $F_c = \frac{m \cdot v^2}{r_{min}}$

3: $F_r = C_{rr} \cdot m \cdot g \cdot \cos(\theta)$

4: $F_{air} = = \frac{1}{2} \cdot \rho \cdot A \cdot C_D * v^2$

To model the subsystem accurately air resistance needs to be considered at each point along the drop, turning the subsystem equation into a differential equation (equation 5)

5: $\frac{d^2x}{dt^2} = g \cdot \sin(\theta) - \frac{k}{m} \cdot (\frac{dx}{dt})^2 - C_{rr} \cdot g \cdot \cos(\theta)$

Where: $k = \frac{1}{2} \cdot \rho \cdot A \cdot C_D$

This equation can be solved using Euler's method, which involves calculating velocities at set time intervals. With this technique the maximum velocity can be established, given the time of the drop and the geometry of the track. With the maximum velocity confirmed, the acceleration of the roller coaster as it traverses the exit curve can be calculated using centripetal force (equation 2).

6: $(v + 1) = v(i) + \Delta t \cdot \frac{d^2x}{dt^2}$

The iterative equation (equation 6) is calculated firstly during the curve after the peak, this considers the changing angle from the horizontal that ranges from 0 to incline of the drop θ. This will model the roller-coaster gradually accelerating as the track curves. The next step is to apply the same iterative equation on the drop itself, the only difference being θ is the constant angle as the track is straight.

The equation will return a discrete array of velocities and a time step of Δt, when given the total time and time spent on the curve as inputs. This velocity profile can then be plotted (Figure 4.1.1.1), showing the acceleration over the initial curve, and the deceleration as the rollercoaster reaches its terminal velocity. The graph is integrated to return the displacement.

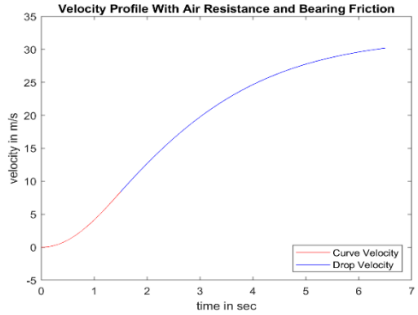


Figure 4.1.1.1

Because the equation is complex and must be solved iteratively the approach was taken to apply the equation to many different inputs of variables θ, radius, time of curve and total time and build a database of different experimental values. This allowed control over how many datapoints were present in the dataset, and the range of each input value.

3.2 Optimisation formulation

To create an enjoyable roller-coaster experience the objective function is to maximise maximum velocity.

3.2.1 Variables

The objective function is defined by the variables shown in table 4.2.1.1.

Table 4.2.1.1: Subsystem 2 Variables

d	The length of the drop
v_c	Velocity after initial curve
G	G Force
t_T	Total time of initial curve and drop
t_c	Time of initial curve
θ	Theta
r	Radius of exit curve

3.2.2 Constraints

Constraints were chosen by analysing the limits of the physical geometry of the track, analysing a database of rollercoasters to evaluate typical values (Figure 4.2.2.1), researching government regulations [15] and safety limits [4]. Constraints were considered at the sampling stage.

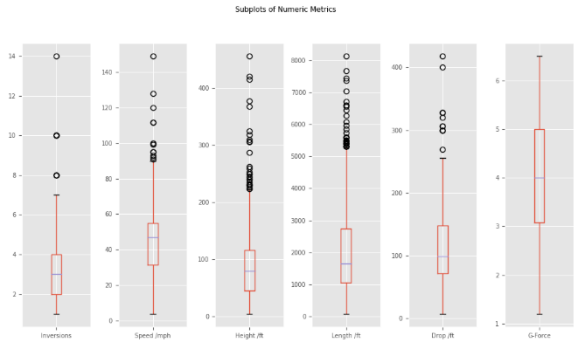


Figure 4.2.2.1

Table 4.2.2.1: Subsystem 2 constraints

g_1	The drop height must be less than 76m	[16]
g_2	The drop height must be greater than 0m	
g_3	The velocity after the initial drop must be less than the terminal velocity.	
g_4	The velocity after the initial drop must be greater than 0ms^{-1}.	
g_5	The G force must not exceed 7 Gs (68.67ms^{-2})	[4]
g_6	The G force must be greater than 1 G (9.81ms^{-2})	
g_7	The time of the initial curve must be greater than the total time	
g_8	The total time must be less than 8000ms	[17]
g_9	The time of the initial curve must be greater than 0s.	
g_{10}	The time of the initial curve must be less than 3000ms	[17]
g_{11}	The incline angle (from the horizontal) must be less than 90°	
g_{12}	The incline angle must be more than 60°	
g_{13}	The radius of the exit curve must be less than 40m	[18]
g_{14}	The radius of the exit curve must be more than 20m	[18]

3.2.3 Assumptions

The subsystem approximates the gradually changing radius of the exit curve as a curve of constant radius. In reality, the maximum acceleration (G force) will occur instantaneously at the point of minimum radius, however in this model the maximum G force will be constant. This will not affect the optimisation as the maximum G force will not exceed safe limits because of the constraints. If the objective was to return the exact geometry of the track, then this would be remodelled with greater accuracy.

Table 4.2.3.1: Subsystem 2 formulated assumptions

h_1	The roller-coaster is at sea-level and at 15 °C and air density is 1.225 kg/m3	[1]
h_2	The frontal area of the roller-coaster is 7m^2	[19]
h_3	The coefficient of drag is 1.2	[20]
h_4	Acceleration due to gravity is 9.81ms^{-2}	[21]
h_5	The mass of the rollercoaster is 500kg	[22]
h_6	The coefficient of friction is returned by the previous sub-system.	

3.2.4 Objective Function

Min: $f(d, v_c\, G, t_T,\ t_c\ ,\ \theta,\ r\ )$

Where: $\boldsymbol{x} = (d, v_c\, G, t_T,\ t_c\ ,\ \theta,\ r\ ) \in X \in \mathbb{R}^n$

$\boldsymbol{p} = (\ \rho,\ A,\ C_D,\ g,\ m,\ C_{rr}\ )$

$h_1(\rho)$: $\rho - 1.1 = 0$
h_2: (A): $A - 7 = 0$
$h_3(C_D)$: $C_D - 1.2 = 0$
$h_4(g)$: $g - 9.81 = 0$
$h_5(m)$: $m - 500 = 0$
$h_6(\mu)$: $C_{rr} - 0.01 = 0$

$g_1(d)$: $d \leq 76$
$g_2(d)$: $-d \leq 0$

$g_3(v_c, g, m, \rho, A, C_D)$: $v_c \leq \sqrt{\frac{2 \cdot g \cdot m}{\rho \cdot A \cdot C_D}}$

$g_4(v_c)$: $-v_c \leq 0$
$g_5(a)$: $G \leq 7$
$g_6(a)$: $-G \leq 1$
$g_7(t_T, t_c)$: $t - t_T \leq 0$
$g_8(t_T)$: $-t_T \leq 8000$
$g_9(t_c)$: $t_c \leq 0$
$g_{10}(t_c)$: $-t_c \leq 3000$
$g_{11}(\theta)$: $\theta \leq 90$
$g_{12}(\theta)$: $-\theta \leq 60$
$g_{13}(r)$: $r \leq 40$
$g_{14}(r)$: $-r \leq 20$

3.3 Modelling approach

3.3.1 Sampling

Equation 6 and 2 were combined into the 'GenerateData' function which would take variables (total time), (time of initial curve), (theta), (radius) and other static values. The function returns the remaining variables of (maximum velocity), (drop distance), (initial slope velocity), (G force). Building the database has $\theta(n^4)$ time complexity for a value of n defined in the script.

Latin hypercube sampling was used to sample the values of the variables between the constraints in order to build the database.

3.3.2 Linear regression

The data was shuffled and normalised between values of 0 and 1, using (equation 7), to ensure the range of values was comparable and avoid any unwanted scaling of the beta values.

Equation 7 $X' = \frac{X - Min(x)}{Min(X) - Max(X)}$

The database was split into features and labels and then each matrix was split again into training and testing data, with a split of 75:25 From here linear regression was implemented on the dataset, using Matlab's 'mvregress' multi-variate linear regression function. This resulted in beta values of (-0.2004, -0.3968, 0.1830, 0.9578, 2559, 0.1553, 0.0483) and an r-squared value of 0.9971, indicating an excellent fit to the data. The residuals were plotted (Figure 4.3.2.1) and their approximate normal distribution indicates the linear model is a good fit to describe the data and the assumption that the data is linear is a good one, however there is a slight skew which indicates the data is heavy tailed, meaning there are several extreme positive or negative residuals.

3.4 Explore the problem space

To gain a better understanding of the problem parameters were plotted independently to analyse how the function behaved as different variables changed. The data point values of total time, max velocity and drop

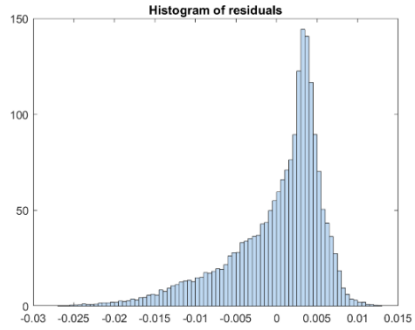


Figure 4.3.2.1

distance (Figure 4.4.1) show a slight curve due to the non-linear effects of considering air resistance.

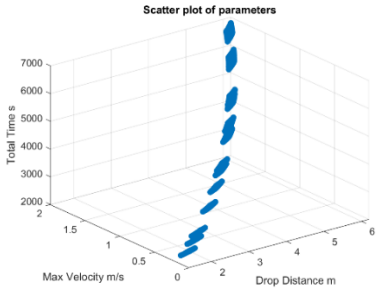


Figure 4.4.1

The monotonicity of the constraints is shown in table 4.4.1. The function is decreasing with regard to d, v_c , $\boldsymbol{G}$, t_T, t_c, θ, r, increasing with regard to t_c and is undefined with regard to $\boldsymbol{G}$ and r. The table shows the problem is well bounded. g_7 and g_{10} both bound t_c but neither can be eliminated because they are both active constraints for different variables. Therefore, the problem cannot be simplified further.

Table 4.4.1: Monotonicity Analysis

Constraint	d	v_c	G	t_T	t_c	θ	r
f	-	-	U	-	+	-	U
g_1	-						
g_2	+						
g_3		-					
g_4		+					
g_5			-				
g_6			+				
g_7				+	-		
g_8				-			
g_9					+		
g_{10}					-		
g_{11}						-	
g_{12}						+	
g_{13}							-
g_{14}							+

3.5 Optimise

3.5.1 Sequential Quadratic Programming

Matlab's 'fmincon' function was used to find the minimum point within the constraints. The SQP algorithm was chosen due to its speed and because the optimisation formulation is constrained and non-linear. The constraints were added as inequality constraints as well as the upper bounds and lower bounds which were defined before building the database. Once the algorithm had been applied to the problem the normalisation was removed and the results are shown below.

Table 4.5.1.1: Sequential Quadratic Programming values

$f(d, v_c\ G, t_T, t_c, \theta, r)$	33.5309 ms^{-1}
d	76 m
v_c	0.5 ms^{-1}
G	6 G
t_T	6990.5 ms
t_c	1998.1 ms
θ	90°
r	40 m
Average Time taken to compute:	0.005522 s

3.5.2 Genetic Algorithm

The genetic algorithm was run several times to verify the results obtained by the SQP algorithm. The values returned by the genetic algorithm are shown in table 4.5.2.1 and vary slightly from the results obtained by SQP. The time taken for the genetic algorithm to run is much larger than the SQP algorithm. Each generation (usually between 100 and 200) is plotted against the best score value and the mean score value. (Figure 4.5.2.1)

Table 4.5.2.1: Genetic Algorithm values

$f(d, v_c\ G, t_T, t_c, \theta, r)$	33.6203 ms^{-1}
d	82.8 m
v_c	0.5 ms^{-1}
G	6 G
t_T	6654 ms
t_c	1930 ms
θ	88.5°
r	38.5 m
Average Time taken to compute:	27.606164 s

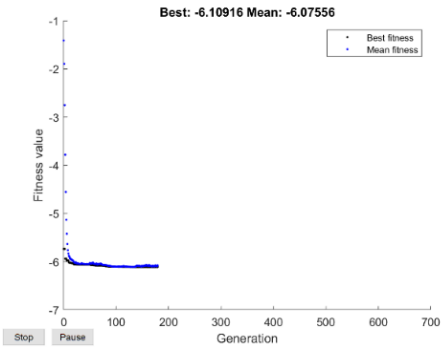


Figure 4.5.2.1

3.5.3 Parametric Analysis

Once the optimisation was complete, a parametric study was conducted by sequentially picking normalised variables to plot against the maximum velocity and the total time over drop. Whilst holding the other variables constant at their mean values. By examining the gradient, the variables can be compared to one another visually and the behaviour of the function can be better understood. Figure 4.5.2.2 shows the effects of varying the normalised angle of incline and normalised radius, and Figure 4.5.2.3 shows the effects of normalised drop distance and normalised time over initial curve. It can be seen that the changes in the normalised radius have a lesser effect on the overall function than the normalised incline. The normalised drop distance has the most effect. This makes sense when considering how the data generation function works; the radius only effects the value of G force, not the maximum velocity and the drop distance would have a larger impact on the objective function due to equation 8.

Equation 8 $\quad Velocity = \frac{distance}{time}$

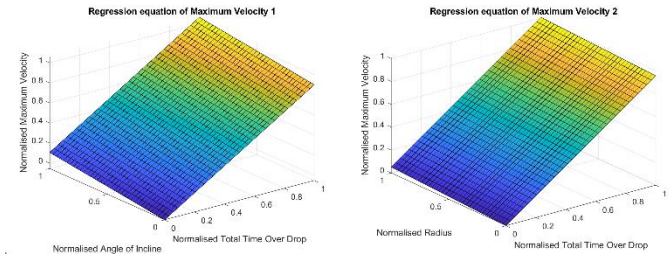


Figure 4.5.2.2

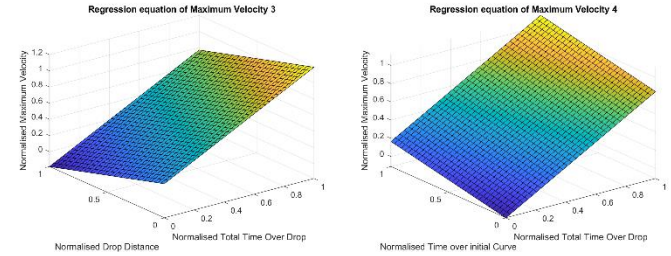


Figure 4.5.2.33

The results were further validated by modelling the geometry of the track in Solidworks, to conduct a visual sanity check.

3.6 Multi objective Problem Reformulation

3.6.1 Pareto search

The problem was reformulated to maximise both the velocity at the end of the drop and the maximum centripetal acceleration at the exit curve. The original optimisation objective was combined with the objective function below, with the same constraints a before.

Min: $f(x) = [f_1(x), \frac{m * f_1(x)^2}{r}]$
Where: $x = (d, v_c\ G, t_T,\ t_c\ ,\ \theta,\ r\ ) \in X \in \mathbb{R}^n$

The normalised pareto set for this formulation is shown in Figure 4.6.1.1. The reason for the discontinuity may be due to concavities on the objective front or may indicate two distinct feasible regions.

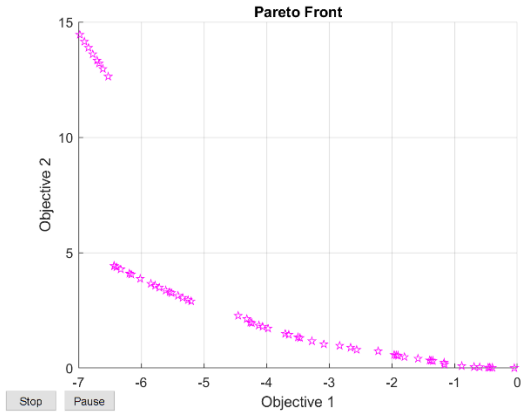


Figure 4.6.1.1

When choosing optimal value from the pareto set, the velocity objective is prioritised because a greater velocity not only results in a more enjoyable experience, but also inputs more energy into the rest of the rollercoaster to allow it to traverse more loops and zero-G rolls, further improving the system level objective function as a whole.

3.6.2 Weighted Multi-Objective Optimisation

To calculate the optimised parameters Matlab's fgoalattain algorithm was used. The goals were both set to 1 and the function of maximum velocity and G force were weighted as 1 and 0.7 respectively and adjusted slightly to constrain the problem. The results are shown in table 4.6.2.1.

Table 4.5.2.1: fgoalattain values

Velocity	31.4552 ms^{-1}
G Force	6.408 G
d	100 m
v_c	0.5 ms^{-1}
G	6.408 G
t_T	6700 ms
t_c	1947 ms
θ	89°
r	22 m
Average Time taken to compute:	0.019552 s

3.7 Discussion

The SQP algorithm gave took the shortest amount of time to run and produced sensible results, verified against existing rollercoasters and research. The model was studied with parametric analysis and run through a genetic algorithm for further verification. Whereas previously the G force variable was constrained manually the multi-objective reformulation set both G force and velocity as objective functions, subsequently improving the formulation. The objective function could be made more accurate by modelling the exit curve as a clothoid curve rather than a curve of constant radius, which can result in large values of jerk.

4. Subsystem 3: The Loop

This subsystem aims to maximise passenger enjoyment by maximising the G-force levels experienced by the passengers, within constraints of safety and feasible geometry. As this method of increasing enjoyment is different to the others it provides more variety to the whole rollercoaster, further increasing enjoyment.

4.1 Optimisation formulation

4.1.1 Negative Null Form

Minimise:

$$f(C, hmax, Rmax) = \sum_{n=1}^{i} (Hi - Gi)$$

Subject to:

$$0 = \frac{C}{2\,g(2\,hmax - 1)} - \theta$$

$$0 = \frac{1}{\theta'} - R$$

$$0 = \frac{2(hmax - h)}{R} + \cos(\theta) - G$$

$$0 \leq Hi - Gi$$

$$0 \leq Rmax - Ri$$

$$0 \leq hmax - h$$

Symbol	Meaning
C	G-Force equation parameter
h (m)	Height of coaster section
hmax (m)	Maximum height of coaster section
R (rad)	Curvature of coaster section
Rmax (rad)	Maximum curvature of coaster section
H (m/s^2)	Safe G-force Levels over time
G (m/s^2)	G-force of coaster over time
g (m/s^2)	Gravity (9.81 m/s^2)
θ (rad)	Angle of coaster section relative to ground

The minimisation problem is formed around using the given parameters to reduce the difference between the summations the plot of Safe G-force levels (H) and the G-Forces incident on the coaster over its journey (G). Practically, this means reducing the area between the 2 when plotted.

The plot of safe G-force levels (below in full) is taken from the ASTM standards for the design of safe amusement attractions [23]

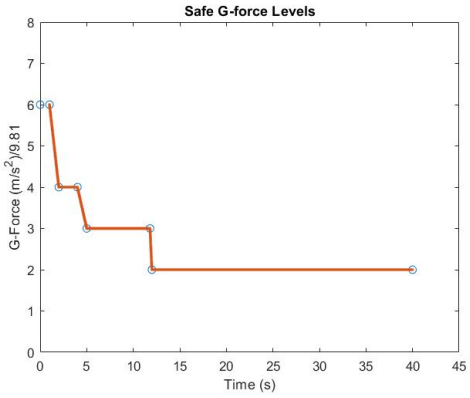


Figure 5.1.1

To find the level of G-force the coaster is exposed to, we first combine expressions of circular acceleration and changing energy levels (from gravitational potential energy to kinetic energy) to find an equation for track angle, θ. The derivative of this is used to find the curvature of the curvature of the track, R. This is used with equations of forces incident on the coaster to find an equation for the G-force. These can be formed into key problem constraints.

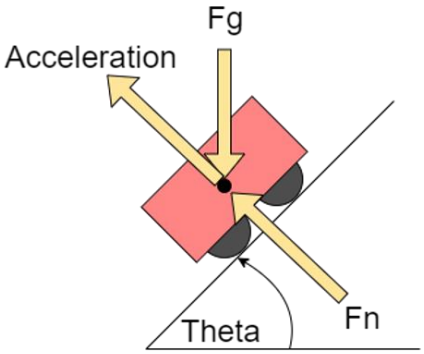


Figure 5.1.2 Free body diagram of a rollercoaster segment

An inequality constraint is provided from obeying the above safety standards. At no point must the G-forces on the coaster exceed safe limits, otherwise the passengers may be harmed. More constraints are provided by geometry. The height and curvature values cannot exceed maximum values outlined below:

Values	Max	Reason	Min	Reason
C	5	G-Forces would surpass safe levels	0	Track would curve downwards
hmax	76	Limit of height provided by subsystem 2	0	Limit of height provided by subsystem 2
Rmax	0.2	Cart would be unable to go around the track	0	Track would curve downwards

4.1.2 Optimisation formulation Assumptions

As with other sections, assumptions are made to simplify the problem:

- Effects of 3-D track geometry are assumed to be negligible
- Energy lost to drag and friction is assumed to be negligible
- Variables to do with the material and cost of the track and the cart are assumed to be irrelevant
- Start and end points have a track angle of 0 radians for simplicity
- The track is on a flat plane. No holes will be dug into the ground, so the minimum height stays at 0.

4.2 Modelling approach

Given the complexity of the calculations involved a computational method would have to be used to model the problem. Since the parameters are set at the beginning of the problem and only first order differential equations are used, Euler's method was chosen to model the problem. This method would break up the track into many smaller sections that can be analysed using a graph to find key information. As this would require large arrays, MATLAB was used to construct, analyse and optimise the model.

In simple terms, the method would lay points of the track by running the previous segment's parameters through these equations:

$$x_{n+1} = x_i + \mathrm{S}*\cos\theta_i$$
$$y_{n+1} = y_i + \mathrm{S}*\sin\theta_i$$
$$\theta_{i+n} = \theta_i + \frac{S * C}{\left(\frac{v0^2}{g}\right) - 2 * y_i}$$

Where S is the length between each new point of track (In this model it was 0.1 m as this was found to retain good accuracy without sacrificing computational speed) and v0 is the speed the coaster starts the loop with, found using energy equations. Once a loop is completed, the x and y arrays can be plotted to see a representation of the loop.

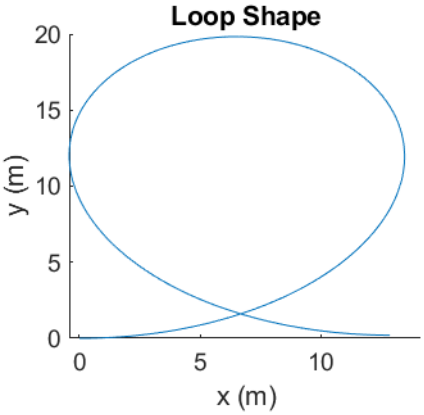


Figure 5.2.1 Loop made with parameters [4.8 35 0.1]

The array of angles can then be used with the constraint functions to create a plot of the G-forces on the curve. Using the y values for each point, the speed along the loop can be found. This with the equations of uniform motion can show the time the coaster reaches each point, and this data used to scale the graph.

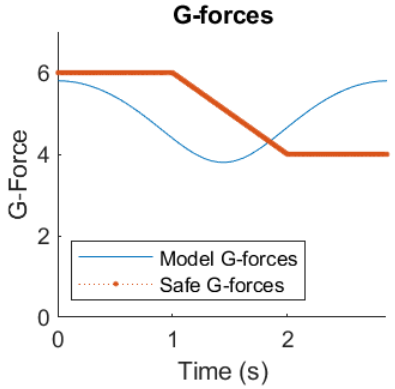


Figure 5.2.2 G-force levels, parameters as above. Note that these parameters are unsuitable as they exceed the safety limits.

The summations of the 2 G-force arrays are found and the difference between them used as a score of the parameter's effectiveness, which we wish to minimise. If the parameters given cause the model to surpass a safety or curvature constraint, a large amount is added to the score, dissuading the parameter's use and creating a 'constraint wall' that can be easily applied to this complex problem.

The key assumptions here are the same as the above with one addition

- The resolution of data that MATLAB gives is large enough.

4.3 Explore the problem space

Due to the complex parts of the problem, such as the first order differentials and nonlinear safety limits, linear monotonicity analysis wouldn't be appropriate here, however the influence of certain parameters can still be observed by trial. Changing the height parameter alone, lengthens the track, so that it can keep the same G-force levels. Increasing the C parameter causes the loop to tighten, impacting the maximum curvature that the loop can take.

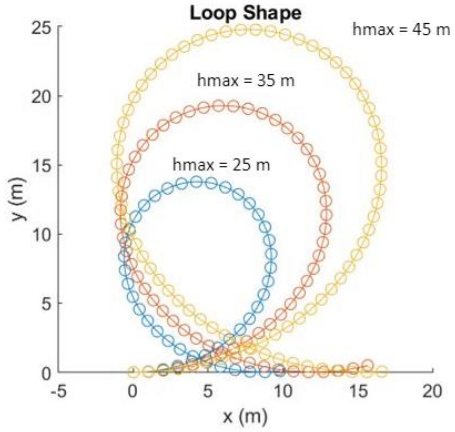


Figure 5.3.1 – Changing the hmax variable

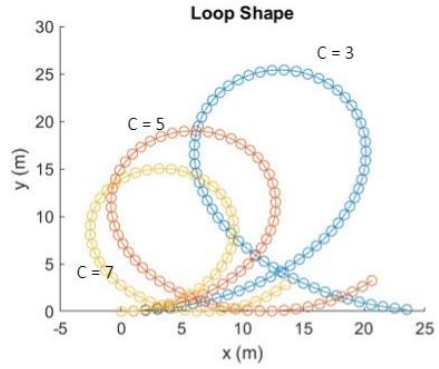


Figure 5.3.2 – Changing the C variable

To explore the problem space further, the model was run multiple times at many different values of hmax (from 10 m to 70 m) and C (from 2 to 5). The scores of the results were collected and used to make a surface plot.

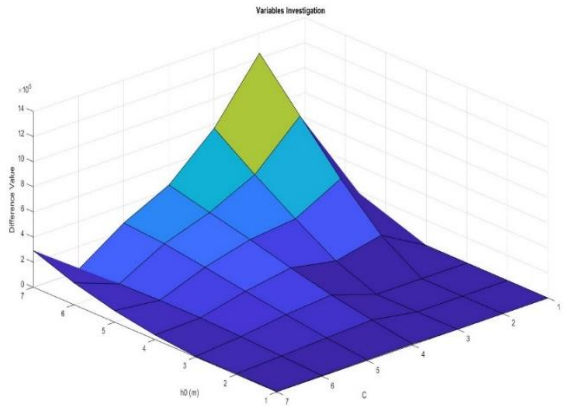
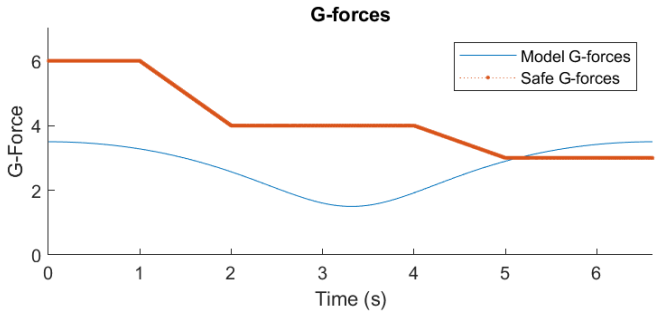


Figure 5.3.3 – hmax vs C vs Score

Due to the large difference in scores, a minimum is hard to spot by eye, but many smaller values are visible and not would be suitable starting points for an optimisation algorithm. Ridges can be seen on the surface. This is due to the step-like safety curve. The maximum of the surface, at C = 2.5 and hmax = 70 m, is a good example of this.



The large kinetic energy from the drop and low level of G-force causes a long, slow loop, which is not completed by the time the safety curve decreases, causing a large score and a ridge along similar values of C. Also notable is that increasing the Rmax value did not change this plot, showing that hmax and C are the key boundaries.

It was interesting to note from a problem space perspective that initial attempts at optimisation only contained the C and hmax variables. This gave a very low score by making the loop extremely small and short. This shows how important the Rmax variable is in getting a feasible solution.

4.4 Optimise

The variable and output of the problem are continuous and the are some constraints on the problem, therefore constrained gradient-based optimisation seems to be the most suitable optimisation algorithm to employ. The most efficient of the ones we have been taught, SPQ, is also packaged into MATLAB's fmincon function, making application straightforward.

Based off earlier problem space analysis a starting point of [1.5, 30, 0.10] is used. It is important to use starting parameters that begin with the G-force under the safe levels as otherwise the optimiser gets stuck in a local minimum and doesn't give a solution that obeys safety constraints. After 7 iterations and 170 runs of the function taking 43 seconds, a solution is found that provides these graphs.

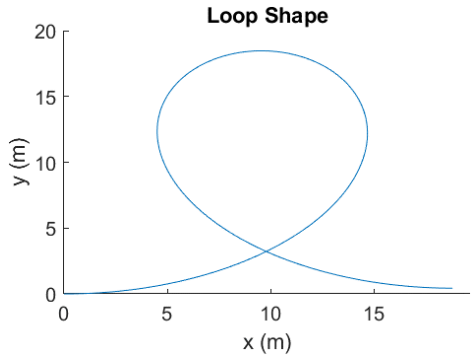


Figure 5.4.1 – Optimal loop structure

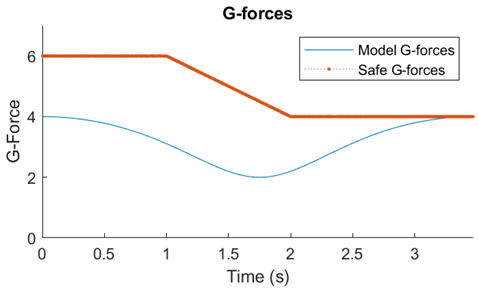


Figure 5.4.2 – Optimal G-force plot

Parameter	Optimised value
C	3
hmax	25.5714 m
Rmax	0.1 rad
Score	**9.015137e+02**

For a full exploration of the optimisation problem different constrained gradient based optimisation algorithms were applied. The large-scale algorithm 'interior-point' was applied and gave a different result in a similar time (51 s). The score of the result was very low at 2.098657e+01, however the resulting G-force plot surpassed safe limits, meaning it is not an appropriate answer. This implies that small-scale algorithms work better and that the model needs to be reassessed to more soundly constrain against minima that cross the safety limits.

4.5 Results, Analysis and Next Steps

The optimised loop would only need to have a start 25.5714 m below the top of the drop. This is feasible and can be added by the rising section of track after the end of subsystem 1. The G-force levels settled just below the second 'step' of the safety curve, giving a maximum level in a very short time. A shorter time would give a better answer but wouldn't be possible due to the maximum curvature constraints.

In all I am happy with this result. Next steps would be to act on the assumptions to make the model more complex and realistic, firstly taking energy losses into account and secondly to ensure this subsection can safely transition to the next. The equation for G-force can also be improved by adding a variable with respect to time, so the G-force plot can decay at the same rate as the safety plot.

5. Subsystem 4: Zero-G Roll

Zero-G rolls are a track section in which passengers experience weightlessness (*airtime*) while performing a 360 degrees inversion. This sensation is achieved as the track follows the same path that the cart would follow in projectile motion, and passengers experience zero G-force. Simultaneously, the track rolls around the *heartline* of passengers to perform the inversion. This subsystem's objective is to maximise the airtime experienced by passengers of this rollercoaster while respecting safety constraints. The *launch track section* (on the left of the *launch point* in figure 6.1**)** determines the *flight path* and is the focus of this optimisation problem. Its curvature was modelled using parametric functions, and feasible designs were sampled using uniform ranges of inputs. This yielded a discrete, unconstrained design space. The optimal solution was found using discrete search algorithms.

Figure 6.1 - a diagram of the cart (red) exiting the *launch track*

5.1 Optimisation formulation

5.1.1 Mathematical formulation

Below is the mathematical representation of this optimisation formulation, where design variable vectors $\boldsymbol{y}$ and $\boldsymbol{\theta}$ and parameter vector $\boldsymbol{s}$ were sampled from parametric functions that computed feasible track shapes. These are included in the constraints and explained in *6.2*.

$$Min \qquad -\boldsymbol{f}(\mathbf{x},\mathbf{p}) = \frac{2\sin(\theta_{max})\sqrt{u^2 - 2gy_{max}}}{g}$$

$$Where \qquad \mathbf{x} = (\mathbf{y},\boldsymbol{\theta}) \in \mathbb{R} \in \mathcal{S}^{Model};\ \mathbf{p} = (\mathbf{s},\ g, u, G_{max})$$

$$Subject\ to$$

$$h_1(\boldsymbol{\theta},\ \mathbf{s}): \qquad \theta_{max} - \sum_{i=1}^{i_{max}} \frac{\partial\theta_i}{\partial s_i}(s_i - s_{i-1}) = 0$$

$$h_2(\mathbf{y},\boldsymbol{\theta},\ \mathbf{s}): \quad y_{max} - \sum_{i=1}^{i_{max}} \sin(\theta_i)\,(s_i - s_{i-1}) = 0$$

$$h_3(\mathbf{y},\boldsymbol{\theta},\ \mathbf{s}): \quad y_{i+1} - y_i - \sin(\theta_i)\,(s_{i+1} - s_i) = 0$$

$h_4(u)$: $u - 36.2389 = 0$
$h_5(g)$: $g - 9.81 = 0$
$h_6(\mathbf{s})$: $s_i - s_{i-1} - 0.1 = 0$

$h_7(G_{max})$: $G_{max} - 6 = 0$
$h_8(\boldsymbol{\theta}, \mathbf{s})$: $\frac{\partial \theta_{max}}{\partial s_{i_{max}}} = 0$
$h_9(\boldsymbol{\theta}, \mathbf{s})$: $\frac{\partial \theta_0}{\partial s_0}, \theta_0 = 0$
$h_{10}(\mathbf{s})$: $s_{total} - \sum_{i=1}^{i_{max}} (s_i - s_{i-1}) = 0$
$h_{11}(\boldsymbol{\theta}, \mathbf{s})$: $\frac{\partial \theta_{i+1}}{\partial s_{i+1}} - \frac{\partial \theta_i}{\partial s_i} - \frac{\partial^2 \theta_i}{\partial {s_i}^2}(s_{i+1} - s_i) = 0$
$h_{12}(\boldsymbol{\theta}, \mathbf{s})$: $\theta_{i+1} - \theta_i - \frac{\partial \theta_i}{\partial s_i}(s_{i+1} - s_i) = 0$
$h_{13}(\boldsymbol{\theta}, \mathbf{s})$: $\frac{\partial^2 \theta_i}{\partial {s_i}^2} - \frac{\partial^2 \theta_{i-1}}{\partial {s_{i-1}}^2} - 0.001 = 0$

$g_1(\boldsymbol{\theta}, \mathbf{y}, \mathbf{s}, \mathrm{u}, \mathrm{g}, \mathrm{G_{max}})$: $\frac{\partial \theta_i}{\partial s_i} - \frac{G_{max} - g cos(\theta_i)}{u^2 - 2gy_i} \leq 0$
$g_2(\boldsymbol{\theta})$: $\theta_{max} - 60 \leq 0$
$g_3(\boldsymbol{\theta})$: $-\theta_{max} - 5 \leq 0$
$g_4(\mathbf{s})$: $s_{total} - 300 \leq 0$
$g_5(\mathbf{s})$: $-s_{total} - 50 \leq 0$
$g_6(\boldsymbol{\theta}, \mathbf{s})$: $\frac{\partial^2 \theta_i}{\partial {s_i}^2} - 0.01 \leq 0$
$g_7(\boldsymbol{\theta}, \mathbf{s})$: $-\frac{\partial^2 \theta_i}{\partial {s_i}^2} - 0.001 \leq 0$
$g_8(\mathbf{f})$: $-f_i \leq 0$

Table 6.1.1.1 Subsystem 4 variables and parameters

f	Maximising *airtime* experienced by passengers	seconds
$\boldsymbol{y}$	Height relative to the starting point	meters
$\boldsymbol{\theta}$	Angle of curvature relative to the *x* axis	degrees
$\boldsymbol{s}$	Track length	meters
g	Acceleration due to gravity	m/s^2
u	Cart's velocity entering the subsystem (output of subsystem 3)	m/s
G_{max}	The maximum vertical upwards G-force all healthy humans can experience for a non-negligible amount of time	G

Table 6.1.1.2 Subsystem-level assumptions to reduce complexity

Energy losses due to friction and air resistance in the launch track section have a negligible impact compared to other design variables and were therefore not considered.
Subsystem 3 track ends on ground level and with zero curvature.
The cart can be modelled as a point mass at the *heartline* of passengers: differences in vertical G-forces experienced by passengers at the front and rear of the cart do not have a significant impact on their safety and enjoyment.
The system can me modelled as two-dimensional: lateral G-forces experienced by passengers during the inversion do not have a significant impact on their safety and enjoyment.

5.1.2 Subsystem diagram

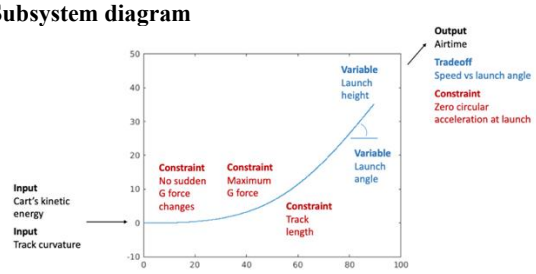


Figure 5.1.1.1 - Summary of subsystem inputs, outputs, design variables and constraints

5.2 Modelling approach

5.2.1 Derivations from first principles

5.2.1.1 Objective function

The time of flight T for a point mass in projectile motion is a function of the vertical component of the object's velocity.

$$T = \frac{2\sin(\theta_{launch})\, v}{g}$$

Considering the loss in velocity due to increasing potential energy as the cart's elevation increases along the launch track, we have:

$$f(\boldsymbol{\theta}, \mathbf{y}) = \frac{2\sin(\theta_{max})\sqrt{u^2 - 2gy_{max}}}{g}$$

Where u is the initial velocity and θ_{max} and y_{max} are the launch height and launch angle respectively.

5.2.1.2 Launch track geometry computation

As curvature is a function of displacement along the track and not of position in the base reference plane, its mathematical formulation in terms of planar *x* and *y* coordinates would involve integrals that cannot be solved analytically. Euler's method was used to compute track shapes numerically. As the length of a segment of the path approaches zero, the change in curvature of the arc with respect to the arc's length approaches the path's partial derivative at that point.

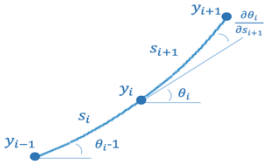


Figure 5.2.1.1 - Segmenting the path to compute track sections

With Euler's method, positions and orientations of points along track sections can be computed numerically and were formulated as equality constraints for the subsystem:

$h_3(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{s})$: $y_{i+1} - y_i - \sin(\theta_i)\,(s_{i+1} - s_i) = 0$
$h_{11}(\boldsymbol{\theta}, \boldsymbol{s})$: $\frac{\partial \theta_{i+1}}{\partial s_{i+1}} - \frac{\partial \theta_i}{\partial s_i} - \frac{\partial^2 \theta_i}{\partial {s_i}^2}(s_{i+1} - s_i) = 0$
$h_{12}(\boldsymbol{\theta}, \boldsymbol{s})$: $\theta_{i+1} - \theta_i - \frac{\partial \theta_i}{\partial s_i}(s_{i+1} - s_i) = 0$

The launch point height and the angle of launch of different track geometries can be computed with the following summations:

$h_1(\boldsymbol{\theta}, \boldsymbol{s})$: $\theta_{max} - \sum_{i=1}^{i_{max}} \frac{\partial \theta_i}{\partial s_i}(s_i - s_{i-1}) = 0$
$h_2(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{s})$: $y_{max} - \sum_{i=1}^{i_{max}} \sin(\theta_i)\,(s_i - s_{i-1}) = 0$

5.2.1.3 Maximum G-force safety constraint

The maximum vertical upwards G-force all healthy humans can experience is 7G (Powell), but a safer 6G was used for this constraint after benchmarking with industry data as the objective of this subsystem is not to maximise G-force. The vertical G-force experienced is a function of acceleration due to gravity, track curvature and cart velocity, the radius of curvature r can be estimated as $\frac{\partial s}{\partial \theta}$ for short segment length s, and velocity is a function of initial velocity u and elevation y:

$G = \frac{v^2}{r} + g cos(\theta_i)$; $\frac{1}{r_i} = \frac{\partial \theta_i}{\partial s_i}$; $v_i = \sqrt{u^2 - 2gy_i}$

Maximum track curvature $\frac{\partial \theta_i}{\partial s_i}$ therefore has an upper limit:

$g_1(\boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{s}, u, g, G_{max})$: $\frac{\partial \theta_i}{\partial s_i} - \frac{G_{max} - g cos(\theta_i)}{u^2 - 2gy_i} \leq 0$

5.2.1.4 G-force variation constraints

As sudden changes in G-force can be detrimental to human health, it was decided to prevent them in this subsystem. G-force is proportional to track curvature, which is defined as the rate of change of the path's orientation with respect to displacement along the path. The derivative of the curvature $\frac{\partial^2 \theta_i}{\partial {s_i}^2}$ must therefore be constant in order to guarantee gradual increases and decreases in G-forces.

At the subsystem level, both the input and output curvature must be zero in order to, respectively, connect with any flat preceding track section and ensure the cart does not have any angular acceleration during 'airtime'. The latter would cause unwanted vertical G-forces to be experienced by all passengers.

$h_8(\boldsymbol{\theta}, \boldsymbol{s})$: $\frac{\partial \theta_{max}}{\partial s_{i=max}} = 0$
$h_9(\boldsymbol{\theta}, \boldsymbol{s})$: $\frac{\partial \theta_0}{\partial s_0}, \theta_0 = 0$

It was decided to design a track path symmetrical around its midpoint to easily locate the point of maximum G-force (*figure 6.2.1.2).* The

curvature increases linearly until the midpoint and decreases at the same rate from the midpoint to the launch point.

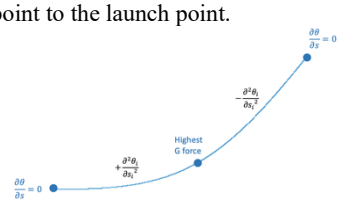


Figure 5.2.1.2 - Geometric visualization of the launch track section

The model for these feasible launch track shapes could not be derived from first principles. As explained in *6.3*, a set of feasible shapes was sampled for a range of $\pm\frac{\partial^2\theta_i}{\partial s_i^2}$ values.

5.2.1.5 Constraints based on assumptions

After benchmarking with existing rollercoasters, it was assumed that track length s_{total} is constrained between 50 and 300 meters in order to limit the number of feasible geometries sampled.

$g_4(\boldsymbol{s})$: $\quad s_{total} - 300 \leq 0$

$g_5(\boldsymbol{s})$: $\quad -s_{total} - 50 \leq 0$

The maximum launch angle was bounded at 60° as, in a real-life situation, from people sitting far from the cart's centre of mass would experience substantial G-forces in the airtime section due to the track 'readjusting' the cart's orientation from θ_{max} at launch point to $-\theta_{max}$ at landing point.

$g_2(\boldsymbol{\theta})$: $\quad \theta_{max} - 60 \leq 0$

The minimum value of the launch angle was set as 5° to filter out feasible 'flight paths' in which the cart orientation of the cart only changes marginally. Although those may yield high flight times, the real-life experience would lack the thrill of feeling weightless as directly facing the sky and then the ground.

$g_3(\boldsymbol{\theta})$: $\quad -\theta_{max} - 5 \leq 0$

5.3 Exploring the problem space

Monotonicity analysis was not appropriate for the constraints derived from first principles as constraint $g_1(\boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{s}, u, g, G_{max})$ is highly nonlinear. The problem space was instead simplified through sampling experiments and by incorporating feasibility limitations (constraints) in the sampling process.

6.3.1 Computing launch track geometries

As the model for launch track shape is parametric, the relationship between the variables $\frac{\partial^2\theta_i}{\partial s_i^2}$ and s_i and the objective function could not be derived analytically. Instead, a MATLAB function was designed to compute a set of feasible geometries for a range of $\pm\frac{\partial^2\theta_i}{\partial s_i^2}$ and s_i values. The launch track is discretised as a sum of short track sections, and the track's length is formulated in h_{10}.

$$h_{10}(\boldsymbol{s}): \quad s_{total} - \sum_{i=1}^{i_{max}} (s_i - s_{i-1}) = 0$$

The function *smoothLaunch* takes $\frac{\partial^2\theta_i}{\partial s_i^2}$ and s_{total} as arguments, computes a path that respects the geometrical constraints outlined in *6.2* and returns the path's launch height y_{max}, launch angle θ_i, and maximum curvature.

5.3.1.1 Launch track geometry sampling parameter tuning

Decreasing the track segment length used to compute consecutive track points is relatively computationally expensive, increasing the running time quadratically due to its use in a nested loop. After performing several experiments and comparing their outputs, this parameter was set to a value that makes outputs converge to three significant figures and makes the design space sampling algorithm executable in under ten minutes in MATLAB Online. This parameter is formulated as an equality constraint as follows.

$h_6(\boldsymbol{s})$: $\quad s_i - s_{i-1} - 0.1 = 0$

5.3.2 Design space sampling & problem simplification

5.3.2.1 Sampling a set of launch track geometries

The function *smoothLaunchSampling* samples a set of launch track geometries by inputting a range of curvature derivatives $\pm\frac{\partial^2\theta_i}{\partial s_i^2}$ and track length s_{total} values into the function *smoothLaunch* outlined in *6.3*.1. Experiments were conducted to test and iteratively improve the range of curvature derivatives $\pm\frac{\partial^2\theta_i}{\partial s_i^2}$ used in this model. Though reality-checking outputs, the following range and step size were selected.

$g_6(\boldsymbol{\theta}, \boldsymbol{s})$: $\quad \frac{\partial^2\theta_i}{\partial s_i^2} - 0.01 \leq 0$

$g_7(\boldsymbol{\theta}, \boldsymbol{s})$: $\quad -\frac{\partial^2\theta_i}{\partial s_i^2} - 0.001 \leq 0$

$h_{13}(\boldsymbol{\theta}, \boldsymbol{s})$: $\quad \frac{\partial^2\theta_i}{\partial s_i^2} - \frac{\partial^2\theta_{i-1}}{\partial s_{i-1}^2} - 0.001 = 0$

For each curvature derivative, *smoothLaunchSampling* computes launch track paths within track length constraints. Points in the feasible track length range at uniform intervals of 0.1 were selected for the sampling as they guaranteed accuracy to at least three significant figures and kept the running time of the sampling algorithm below ten minutes in MATLAB online.

$h_4(\boldsymbol{s})$: $\quad s_{total_i} - s_{total_{i-1}} - 0.1 = 0$

6.3.2.2 Incorporating constraints in the sampling process to simplify the design space

Design spaces with design variables launch angle θ_{max}, curvature derivative $\frac{\partial^2\theta_i}{\partial s_i^2}$, launch height y_{max} and track length s_{total} approximated as continuous functions (figure 6.3.2.1) are highly convex as two of the variables are angular (the derivative of curvature, and the resulting launch angle) and produce noisy overlapping sinusoidal surfaces.

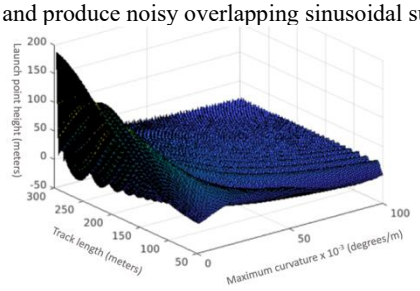


Figure 5.3.2.1 - Early surface plot to investigate the relationship between the sampling algorithm inputs and outputs

Although launch angle θ_{max} could be bounded to respect the problem's constraints, in this model where discretised ranges of track length s_{total} and curvature derivative $\frac{\partial^2\theta_i}{\partial s_i^2}$ were used to compute a partially bounded launch angle θ_{max} and launch height y_{max} design space (figure 6.3.2.1), the same launch angle can be reached through numerous different launch track designs with different combinations of track length and curvature derivative. Some of these variable combinations produced launch track designs with one or more loops – these are not realistically acceptable, and produce numerous non-viable points (and subsequently local minima) in this design space.

Smoothing with a surrogate function would not have been appropriate at this stage as viable points in figure 6.3.2.1 are not necessarily represented by maxima or minima, and may lie anywhere in the design space's periodic surfaces. This numerical noise is important and must be removed to optimise for track shapes that are actually feasible. It was therefore decided to apply the G force safety constraint $g_1(\boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{s}, u, g, G_{max})$ as part of the sampling algorithm. This filtered out all dangerous track geometries that had high maximum curvature derivatives, including any track design whose maximum curvature derivative is so high that it formed one or more loops. Figure 6.3.2.2 shows the relationships between launch heights, launch angles and maximum curvature of the remaining 535 viable track geometries.

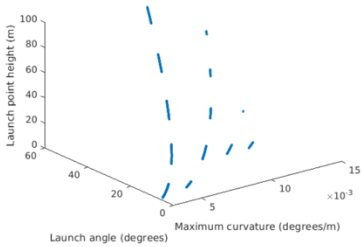


Figure 5.3.2.2 - Feasible launch track geometries

5.3.3 Objective function design space

Using feasible pairs of values of launch angles and their corresponding launch point heights, feasible objective function values could be computed and plotted against the two design variables, launch angle θ_{max} and launch height y_{max} (figure 6.3.3.1). This plot further filters out launch track geometries in which the cart wouldn't have enough initial velocity to reach the launch point. This constraint is formulated as:

$g_8(\boldsymbol{f})$: $\quad -f_i \leq 0$

f_{491} to f_{535} violate this constraint as launch height is too high for the cart to reach the launch point, which makes the objective function return a complex imaginary *airtime* value. These designs were therefore ignored in optimisation by setting an upper bound on the index of f as 490.

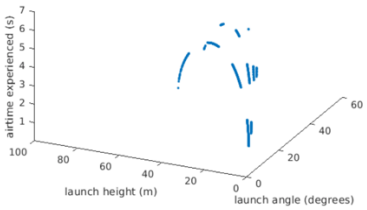


Figure 5.3.3.1 - This subsystem's discrete design space, visualised as a maximisation problem

5.4 Optimisation

This design space is discrete and unconstrained as all constraints were applied during the modelling and sampling process. The relatively low amount of feasible designs makes computational efficiency of the optimisation algorithm negligible. Optimisation methods with integer constraints are appropriate to solve for an index within the set of possible solutions.

5.4.1 Integer Genetic Algorithm optimisation

A genetic algorithm solver was set up with the objective function as the fitness function. The penalty function corresponds to the output of the objective function with the current iteration's indexed launch angle and launch height.

6.4.1.1 Parameter setting

The number of variables was set as 1; equality and inequality constraints were left blank as they were all applied manually in the problem simplification process; upper and lower bounds for the indexes were set as 1 and 490 respectively to prevent out-of-range indexing in the fitness function; and finally the index variable was defined to be integer. Other algorithm properties were left as default.

6.4.1.2 Optimisation results

After repeated testing, the algorithm identifies the best penalty value to the in 1-22 iterations, continues to run until the best penalty value hasn't changed for 50 generations. The process takes 0.10-0.15 seconds. Performance is satisfactory for this scope.

$$f_{min} = -5.3257$$
$$i_{min} = 245$$

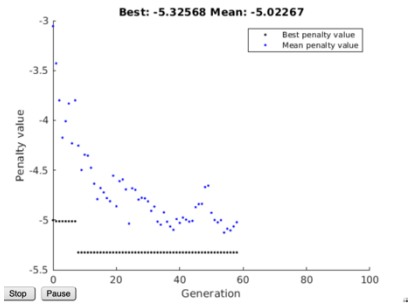


Figure 5.4.1.1 - A plot of the best and mean penalty values found in each generation in an execution of the genetic algorithm

5.4.2 Pattern Search optimisation

Patten Search was selected to test an alternative algorithm. The algorithm balances global and local search. Pattern Search was chosen to test a local search algorithm's performance in solving for an index within a set of solutions. As indexing errors were encountered for very small mesh sizes, however, the algorithm did not perform consistently.

6.4.1.1 Parameter setting

Numerous starting points were tested and yielded different results. Equality and inequality constraints were left blank as they were all applied manually in the problem simplification process, and upper and lower bounds for the indexes were set as 1 and 490 respectively. Other algorithm properties were left as default. The polling method *MADSPositiveBasis2N* converged in fewest iterations and with the lowest mesh sizes.

6.4.1.2 Optimisation results

As the algorithm stops running after few iterations (8 to 18 depending on initial mesh size and polling method) performance was highly dependent on the starting point and mesh parameters (size and scaling factors). Below are the most frequent results.

$$f_{min} = -5.01286, -4.3738$$

5.5 Discussion

The majority of the challenges faced in this subsystem optimisation were in designing experiments to create a model of a function that could not be solved analytically. Moreover, as the initial models were highly non-linear, it was decided apply the constraints as filters throughout the modelling process rather that in the optimisation algorithm. The drawback of this approach is that post-optimal sensitivity analysis cannot be performed mathematically.

Discrete optimisation algorithms yielded feasible designs in a very low computation time. Two algorithms were tested to solve for the index which corresponds to the optimal value in the feasible design variable values vectors. The Genetic Algorithm was efficient, precise and reliable as its global search approach makes it cover the entire discrete design space, which in this case was highly nonlinear as shown by the lines interpolated between points in figure 6.4.2.1.

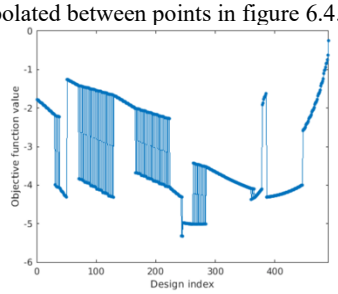


Figure 5.4.2.1 - Optimisation algorithm design space

A local search algorithm (Pattern Search) was much less effective for this application: when converging to a local minimum, the algorithm would decrease the mesh size until it tried to poll non-integer points, hence creating an indexing error. As indices are simply positions in a dataset, this highly nonlinear and non-convex design space could have been simplified using a sorting algorithm and the optimisation itself would have become trivial. The optimal launch track shape (figure 6.4.2.2) has the following design variable values:

$$Launch\ angle = 31.34^o$$
$$Launch\ height = 22.50m$$
$$Track\ length = 85.40m$$

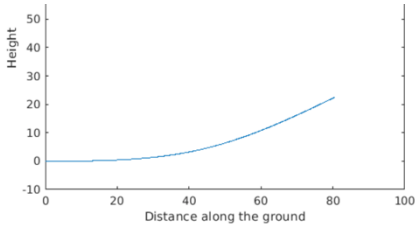


Figure 5.4.2.2 - Optimal launch track sub-subsystem shape

Some assumptions were made to simplify the problem due to the inherent complexity of this modelling process. Optimising for a real-life design, however, would require different geometrical constraints set by the consideration of energy losses along the track and variability in input parameters due to uncertainty in passenger size and weight. Moreover, G-forces in three dimensions should be considered to design the airtime inversion around passenger's heartlines.

6. Conclusion

This rollercoaster has industry-leading performance in the enjoyment factors identified in our research: velocity, G-force intensity and zero-G airtime. The track subsystems were optimised using three different approaches, each with different qualities: *subsystem 2* benchmarks against industry records, *subsystem 3* design according desired G-force profiles and *subsystem 4* considers curvature changes to prevent dangerous sudden G-force changes. To design for a real-world coaster, the methods used in our track subsystems can be combined to optimally design a wide variety of track sections.

Further improvements should include considering uncertainty in initial conditions (passenger number, weight, seating arrangement) to design a robust system, and optimise for reliability to avoid all risks of constraint violation.

7. Appendix A: Nomenclature

7.1 Cart Subsystem

w	Width of the wheels (m)
W_p	Load per wheel from the people in the cart (N)
W_c	Load per wheel from the cart itself (N)
ρ	Density of the wheel material (N/kg^3)
g	Acceleration due to gravity (m/s^2)
d_o	Outer diameter of the wheel (m)
d_i	Inner diameter of the wheel (m)
σ	Poisson ratio of the wheel
σ'	Poisson ratio of the tracks (steel)
E	Elastic (Young's) Modulus of the wheel (Pa)
E'	Elastic Modulus of the tracks (Pa)
F	Rolling resistance (N)
W	Total load per wheel
z	Wheel sinkage
W_w	Load per wheel due to the mass of the wheel (N)

7.2 Drop Subsystem

d	The length of the drop (m)
v_c	Velocity after initial curve (ms^{-1})
G	G Force
t_T	Total time of initial curve and drop (s)
t_c	Time of initial curve (s)
θ	Theta (degrees)
r	Radius of exit curve 9m)
ρ_{air}	Density of air (kg/m^3)
A	Frontal area of the roller-coaster (m^2)
C_D	Coefficient of drag
C_{rr}	Coefficient of friction
g	Acceleration due to gravity (m/s^2)
m	Mass of the rollercoaster (kg)

7.3 Loop Subsystem

C	G-Force equation parameter
h	Height of coaster section (m)
$hmax$	Maximum height of coaster section (m)
R	Curvature of coaster section (rad)
$Rmax$	Maximum curvature of coaster section (rad)
H	Safe G-force Levels over time (m/s^2)
G	G-force of coaster over time (m/s^2)

7.4 Zero-G Roll Subsystem

y	Height relative to the starting point (m)
θ	Angle of curvature relative to the *x* axis (°)
s	Track length (m)
u	Cart's velocity entering the subsystem (m/s)
G_{max}	The maximum vertical upwards G-force all healthy humans can experience for a non-negligible amount of time (m/s^2)

8. Bibliography

[1] thoughtco. [Online]. Available: https://www.thoughtco.com/density-of-air-at-stp-607546.

[2] T. Harris, "., 2008. How roller coasters work. Science. howstuffworks. com.," [Online].

[3] S. Griffiths, "What makes an exciting rollercoaster?," *Ingenia,* no. 66, March 2016.

[4] J. P. a. R. P. Powell, ""Passenger stability within moving railway vehicles: limits on maximum longitudinal acceleration." Urban Rail Transit 1.2 (2015): 95-103.".

[5] Wikipedia, the free encyclopedia, "Rolling Resistance," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Rolling_resistance. [Accessed 3 December 2019].

[6] Technica Total Wheel Solutions, "Introduction to Roller Coaster Wheels," 2019. [Online]. Available: https://www.technicawheels.co.uk/intro-to-roller-coaster-wheels. [Accessed December 2019].

[7] CES EduPack 2019, *Material Properties,* CES, 2019.

[8] A. Simonis and C. Schindler, "Measuring the wheel-rail forces of a rollercoaster," *Journal of Sensors and Sensor Systems,* vol. 7, no. 469, 10 September 2018.

[9] S. Pheasant, Bodyspace, Anthropometry, Ergonomics and the Design of Work, vol. 2nd Ed., London: Taylor & Francis, 1998.

[10] Nick, “Coaster Physics Calculations,” Coaster101, 16 November 2010. [Online]. Available: https://www.coaster101.com/2010/11/16/coasters101-coaster-physics-calculations/. [Accessed 3 December 2019].

[11] Cedar Point, “Guest Assistance Guide,” Peanuts Worldwide LLC, Sandusky, OH 44870-5259, 2017.

[12] Wikipedia, “Carbon Steel,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Carbon_steel. [Accessed 3 December 2019].

[13] L. D. Landau and E. M. Lifshitz, Theory of Elasticity, vol. 7, U. A. o. S. Institute of Physical Problems, Ed., Pergamon Press, 1970.

[14] W. Bessette, “Materials Used in Roller Coasters,” AZO Materials, 24 April 2015. [Online]. Available: https://www.azom.com/article.aspx?ArticleID=11958. [Accessed 3 December 2019].

[15] A. Väisänen, “"Design of Roller Coasters." (2018).”.

[16] Viegas, “Development of Computational Models of Roller Coaster Vehicles and Occupants for Injury Analysis. Diss. MSc Thesis, Instituto Superior Técnico, University of Lisbon, Portugal,,” 2016.

[17] J. a. J. A. Pombo, “Modelling tracks for roller coaster dynamics. International journal of vehicle design 45.4 (2007): 470-500.”.

[18] J. F. a. R. H. G. Mares, “Full range of motion roller coaster. U.S. Patent No. 5,791,254.,” 11 Aug. 1998..

[19] V. G. a. V. A. Gnezdilov, “1995. Roller coaster. U.S. Patent 5,463,962.”.

[20] K. Hunt. [Online]. Available: https://web.wpi.edu/Pubs/ETD/Available/etd-042318-205416/unrestricted/Design_Analysis_of_Roller_Coasters_MS_Thesis_KHunt.pdf.

[21] A. H. Cook, “"The absolute determination of the acceleration due to gravity." Metrologia 1 (1965): 84-114.”.

[22] J. Barriga, “"Rollercoaster-Energy Transformation." (2014).”.

[23] ASTM Committee F24, “ASTM Standards on Amusement Rides and Devices,” ASTM International. [Online]. [Accessed November 2019].

[24] J. a. J. A. ". t. f. r. c. d. Pombo, *International journal of vehicle design 45.4 (2007): 470-500.*.