A Real-time Simulator for the Sport of Skeleton

S.G. Shankar

TU Delft

A Real-time Simulator for the Sport of Skeleton

by

S.G. Shankar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday, December 18th, 2019 at 12:15 PM.

Student number: 4734238
Project duration: June, 2019 – December, 2019
Thesis committee:

Prof. Dr. Ir. A. Schwab,	TU Delft, supervisor
Prof. Dr. Ir. A. Seth,	TU Delft
Prof. Dr. Ir. B. Shyrokau,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

This thesis is the result of a years time spent learning, first about the sport of Skeleton followed by real-time simulation methods for mechanical systems. I have grown a lot during this time both as an engineer and as a person. I would like to thank the people who made it an enjoyable experience. First, my parents, who continue to shower me with love and support from thousands of miles away. Thank you and I Love you! I would not be who I am today if it weren't for the two of you.

I would like to thank my supervisor, Dr. ir. Arend Schwab for all the insightful discussions and more so for the enthusiasm shown towards my work. At times, it exceeded even my own. If it was not for your guidance in simplifying the models formulated here, I doubt that I would have made as much progress as I have.

To my friends and colleagues from BMD, we have shared a journey for more than two years now and I will always be grateful to know such wonderful people and the motivation you have given me over the years. Camilo, our love for sports brought us together and it still does, I hope that our friendship continues to grow as it has done so far. To Mathijs and Prajish, for the all discussions and putting up with the nonsensical ways of how I think sometimes. Cathy, Jens, Chris and Ion, thank you for all the fun and serious time spent together. You are the reason I had any sort of a social life in Delft. Last but not least, the library lunch group! Thank you Aneesh, Krishna, Ashwin and Rahul, you guys were great distractions as I toiled away these past few months.

To everyone at Excelsior '20, Schiedam, when I arrived here in August 2017, I believed my cricket playing days were over. All of you have shown me nothing but kindness and support as I continue to play the sport that I love. Thank you! I will try my best to stay with the club for as many seasons as possible.

I end with family and friends from India. I am the worst at keeping in touch with people whom I do not meet. This has never been more evident than in the past two years. Thank you for making the time and putting in the efforts to reach out to me.

S.G. Shankar
Delft, December 2019

Abstract

The track bound sliding sport of Skeleton was permanently added to Winter Olympics programme in 2002. This has led to increased interest in the sport. Engineering has already proved to be a vital contributor to improved performance in the related sport of Bobsleighing. We hope that engineering can do the same for Skeleton.

This report describes an attempt at developing a platform to be used as a real-time training simulator for the sport of Skeleton. For a multitude of reasons athletes are, on average limited to a total of two hours of practice and competitive on-track time in any given year. When compared with time spent practising and in competition in most sports, this is extremely low. It is hypothesized that a simulator can augment track time by providing a realistic environment to practise in, even when access to a track is not available.

This work is guided by simulators that have been developed for Bobsleighing. The main components are the models to describe the dynamics of the sport, an input method and visualization of the simulation. The main considerations for the dynamic model are of the track surface, the sled and contact between sled and track surface. These models lead to a system of equations which when solved provide accelerations and contact forces. The accelerations are integrated over a fixed time interval to determine changes in velocities, position and orientation. The position and orientation obtained after the integration is passed on to a game engine which provides the user with real-time visual output of the position and orientation along a digitally recreated track surface. A video game controller was chosen to serve as the input device. It has two joysticks, which can be mapped so as to mimic the forces applied by an athlete.

A number of descents were performed using this platform both at real-time speed and at a slower speed to give the user, unfamiliar with the sport, a better chance to steer the sled. We were able to consistently reach the exit of curve 2 in real-time speed and curve 4 at the slower play speed before failure of the simulation. In most cases the algorithm used here proves to take lesser time for computation than the chosen integration time step, which is a great sign for future development as we did not make any attempts to optimise its computation time.

We made an attempt at validation using time elapsed to traverse a certain distance and the sum of magnitude of Lagrangian multipliers. We had poor results with the time elapsed comparison, with simulated runs being 15% slower than competitive descents. While the sum of Lagrangian multipliers showed good relation to expected behaviour. This first attempt was reasonably successful, and we believe that the lessons learnt from this work has brought us one step closer to realizing a training simulator that can be useful to Skeleton athletes.

Contents

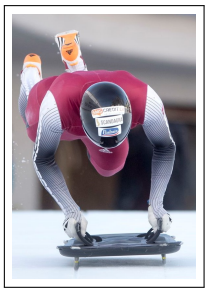
List of Figures

List of Tables

1 Introduction

Skeleton along with Bobsleighing (or Bobsledding) and Luge, belongs to the category of track-bound winter sliding sports. In Skeleton, a single athlete begins a run by sprinting while pushing their sled besides them. Once they have reached top speed, they jump onto the sled and assume a head first, prone position, this motion is referred to as “loading”. The sprinting and loading phase together are referred to as the “push-start" phase. The remainder of a run is a gravity-powered descent down a track that is around 1.5 km long with an average vertical gradient of 10%. The three phases are shown below in Fig. 1.1. A single run lasts close to 1 minute, with speeds exceeding 130 km/hr and athletes experiencing forces up to 5 times that of gravity. The winner of a competition has the lowest cumulative time over two or four runs and often winning margins are fractions of a second. More information about the sport, sleds used and steering methods can be found in Appendix A.

(a) Sprint start



(b) Loading

(c) Descent

Figure 1.1: The different phases of a Skeleton run. Figures a and b with the sprint and loading comprise the push-start phase. Figure c shows an athlete during the gravity powered descent down a track

1.1. Motivation

Even though the majority of the descent is gravity-powered, athletes must perform accurate steering manoeuvres to achieve good finish times. Previous analyses have shown that steering errors can lead to substantial

time losses and have also shown a positive correlation between competitive experience and performance [1–5]. Hence practising steering skills and spending as much time as possible on the ice is critical, but in reality is severely lacking. First, track access is limited geographically, with only 5 out of the functioning 16 tracks located outside Europe[1]. The Netherlands does not have a track. Second, tracks are closed for athletes during the summer. They are repurposed to serve as promotion centres using wheeled sleds with tourists as the riders. Third, there are a limited number of descent slots available at a track per day and due to the forces experienced during a descent, the number of runs any individual athlete can make in a single day is restricted. The above factors combine together to limit athletes to an average of two hours of on-track sliding time over a competitive season [6]. A season is active for roughly half the year. For the remainder of the year, athletes are consigned to strength and conditioning workouts and push-start practice[2]. This is in stark contrast to many sports where an athlete may spend several hours practising requisite skills in just a single day.

We hypothesize that a real-time training simulator will help to overcome these disadvantages. Also, given the high speeds and G-forces involved, more often than not an athlete must steer conservatively due to the risk of injury. A simulator could hypothetically give athletes the opportunity to try a wider range of steering techniques and strategies before attempting to execute them on track [7]. While there have been attempts at developing real-time simulation capabilities for the related sport of Bobsleighing, we are not aware of such attempts for Skeleton [6–9]. These Bobsleigh simulators served as a guide for a majority of the work described in this thesis.

1.2. Thesis Outline

This thesis describes a platform for real-time simulation of Skeleton. Along with the published information regarding Bobsleigh simulators, a second guiding theme was simplicity, be it in the track reconstruction or formulation of the dynamic model for the sport.
Being a track-bound sport, creating an accurate representation of the track surface is essential. This was achieved for the Olympic track at Park City, Utah, The USA using construction data and video footage of a descent, this serves as the second chapter. The third chapter describes the dynamic model used for the simulation. This chapter also includes details of the time-stepping algorithm used. The work presented here is only concerned with the simulation of the descent once the athlete has loaded on to the sled at the end of the push-start phase. Chapter 4 explains the simulation set-up including input and output methods and devices used. It further includes estimation of the various parameters. Chapter 5 discusses the results of simulated runs along with an attempt to validate the simulator. We end with recommendations for future work and conclusions in Chapter 6. The appendices contain a more involved description of the sport of Skeleton, Bezier curves used for creating the track model in Chapter 2.

[1] `https://en.wikipedia.org/wiki/List_of_bobsleigh,_luge,_and_skeleton_tracks` Retrieved: 01/12/2019
[2] Personal correspondence with Akwasi Frimpong, a Skeleton athlete

2

Track Model

The track surface plays a vital role in realizing a realistic simulation of Skeleton. An analytical or numerical approximation of the track is needed for the physics simulation. Additonally, a graphical representation of the surface is needed for the visualization of a descent. The following chapter describes the process followed to generate a numerical track surface model, used for both the physics simulation and its visualization.

The method to reconstruct the 3D track surface is based on the approaches used by Rempfler and Glocker [6] and Braghin et al. [9, 10], who have developed simulations for Bobsleighing. In brief, one begins by first approximating the Track Profile (or Centre) Line in 3D space. The second step is to assemble curves that describe the geometry of the ice surface transverse to the Track Profile Line, henceforth referred to as transverse cross-sections, along the Track Profile Line. The third step is to interpolate between these transverse cross-sections. At the end of this step, one obtains a distribution of points in 3D space. The fourth step involves connecting the points, 3 at a time to obtain a mesh of triangles. This step is known as Triangulation. The final step is a parametrization of this triangular mesh to obtain useful quantities for use in the dynamic model.

The aforementioned studies had complete access to construction and planning data for the tracks they use (Whistler, Canada and Cessana Pariol, Italy respectively). This contains data on the Track Profile Line and transverse cross sections as one moves down the track. The Netherlands does not have its own track and requests for this data from track authorities of several tracks were not successful. Only information regarding the Track Profile Line for the Olympic Track at Park City, Utah, The USA was obtained. To overcome this deficit, video footage of a descent down the track is used to approximate the transverse cross-sections.

2.1. Method

2.1.1. Track Profile Line

The reference frame $\mathcal{N}$ used here is represented by a right handed orthogonal coordinate system XYZ with basis vectors $\hat{\boldsymbol{e}}_x^{\mathcal{N}}, \hat{\boldsymbol{e}}_y^{\mathcal{N}}, \hat{\boldsymbol{e}}_z^{\mathcal{N}}$. Gravity acts in the $-\hat{\boldsymbol{e}}_z^{\mathcal{N}}$ direction. The Track Profile Line is first determined in the horizontal XY plane. In the available data there are two sources. The first is a plot of the right and left extremes of the track surface shown in Fig. 2.1 & 2.2 and is hereafter referred to as “Graph Source". The second shows among others, horizontal curvature and elevation of the track profile line as a function of distance along the track. It is shown in Fig. 2.3 and 2.4, it will be referred to as “Numerical Source".

Graph Source

The graph has two lines, the left side and right side extremes of the track surface, along with cross bars every two meters measured along track direction, these can be seen in Fig. 2.2. By taking the mean between the two extreme lines one obtains an approximation of the Track Profile Line. The line obtained is limited in usefulness, for example it is hard to determine the exact point of transition between straights and curves and neither does it contain elevation data. The importance of this will be made clear in subsequent sections. It is however, used to overcome a deficit of the line obtained from the Numerical Source, which will also be explained later.

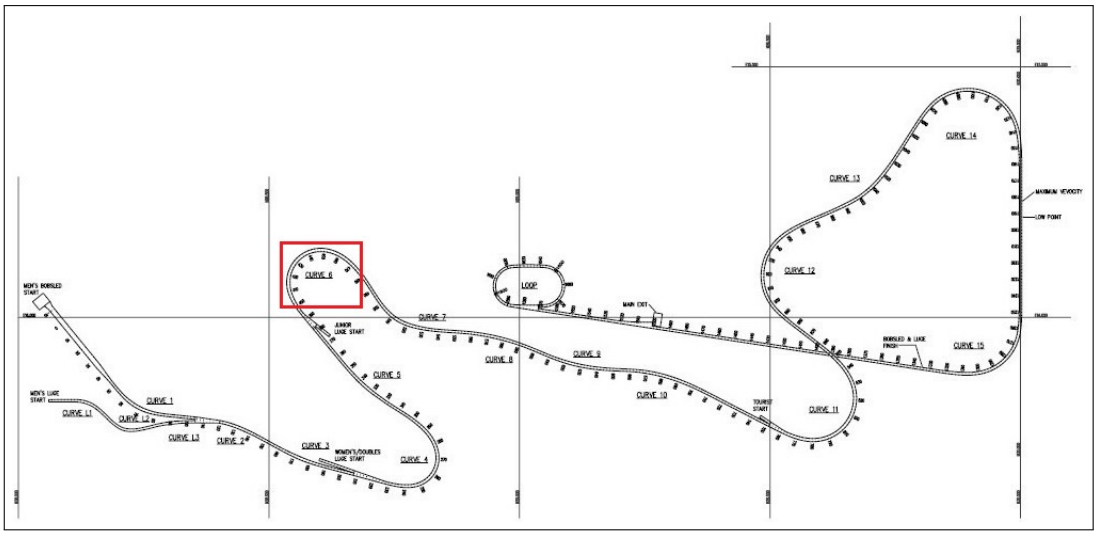


Figure 2.1: Graph Source for Track Profile Line at Park City, Utah. Curve 6 is highlighted by the red square, an enlarged view of this area is shown in Fig. 2.2 below.

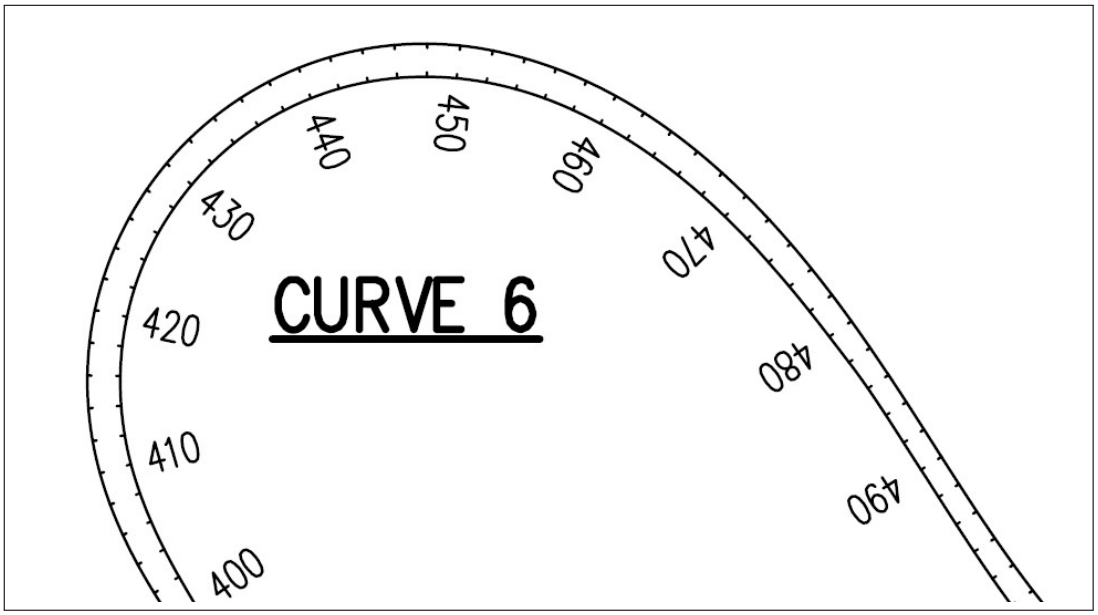


Figure 2.2: Graph Source - Enlarged view of curve 6. The numbers are distance measured along the track with cross-bars every 2 m

Numerical Source

The graphs here contain two important pieces of information used to reconstruct the Track Profile Line, they are the horizontal curvature and elevation of the track surface at Track Profile Line and can be seen in Fig. 2.4. The data from the horizontal curvature and slope is used to divide the line into straight and curved segments as shown in Fig. 2.5. The Track Profile Line was divided into 34 segments. The segments are constructed individually and for a straight segment of the track, a straight line of the required distance is created. While for each curved segment, the data contains the length of the segment and radius of curvature, which are used to create the segment. Starting with the first segment, the angle between the tangent to the line at its last point and X axis is determined. The succeeding segment is first rotated by this angle and then translated to the last point of the preceding segment. This process is repeated for all the segments to form the complete Track Profile Line.

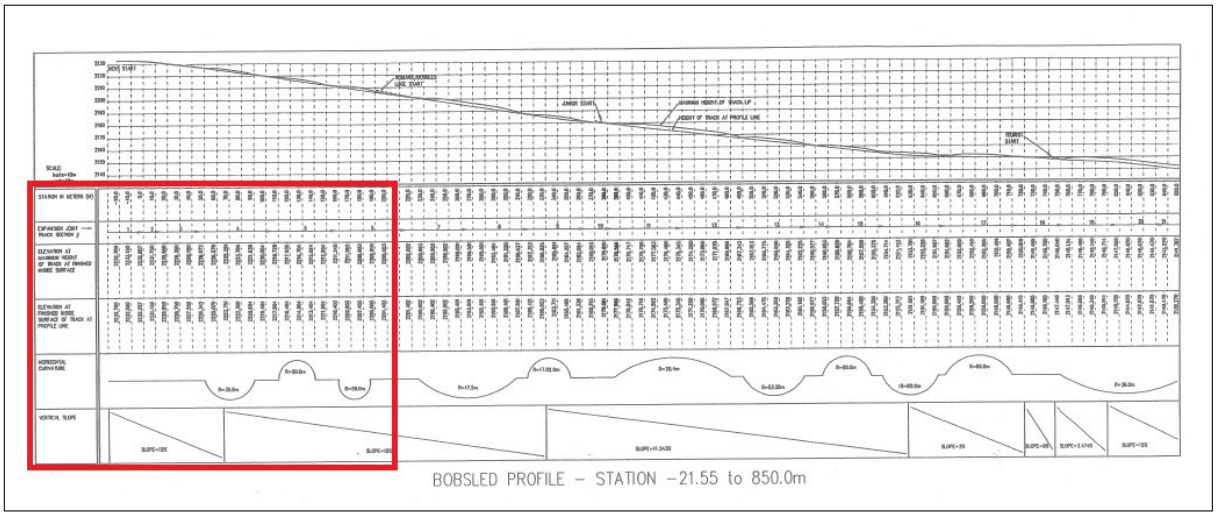


Figure 2.3: Numerical Source for Track Profile Line at Park City, Utah, The USA. Region highlighted by red square is shown in Fig 2.4 below. It contains the data for the first 200 m of the track.

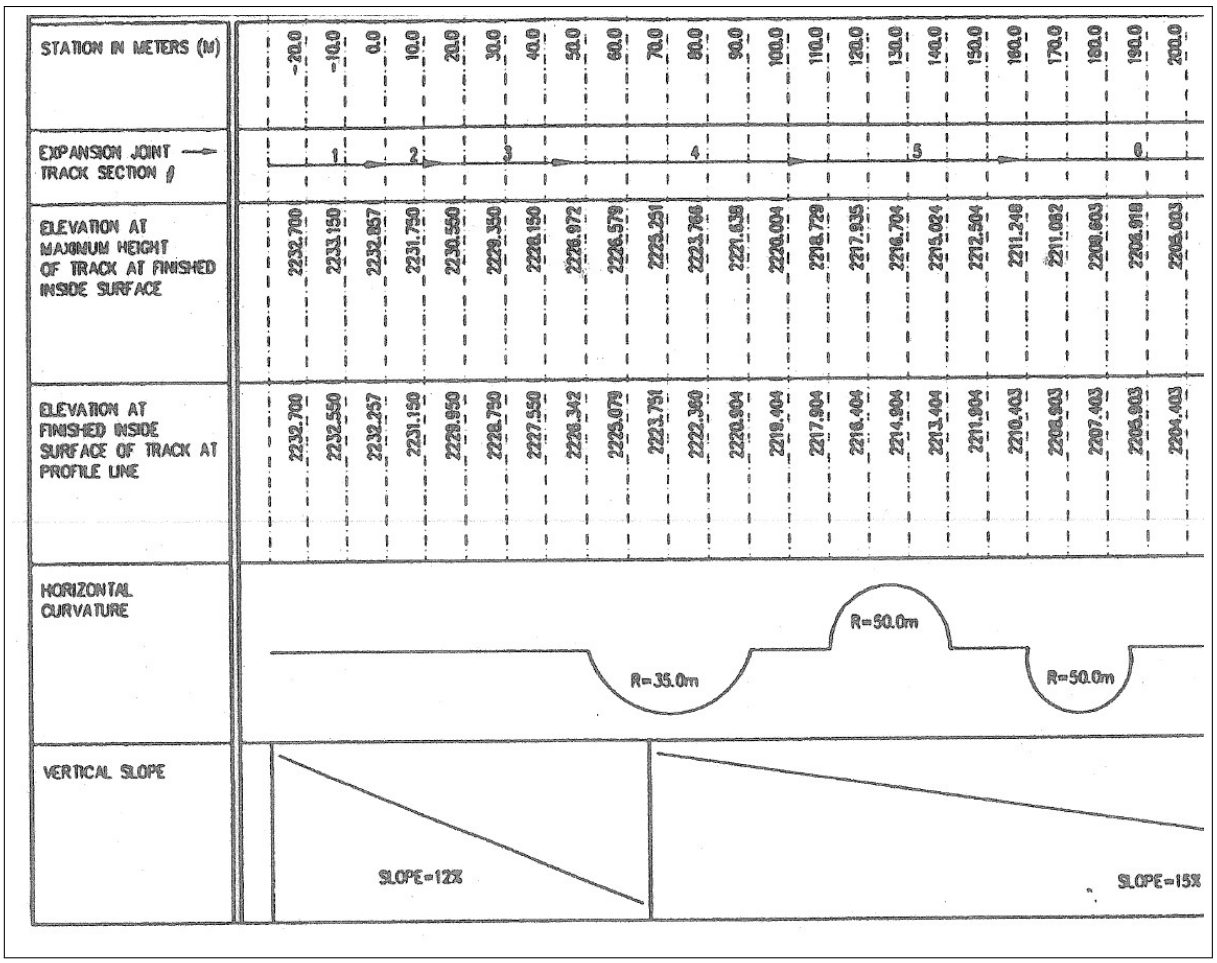


Figure 2.4: Numerical Source - Enlarged view of the data for first 200 m of the track at Park City, Utah, The USA

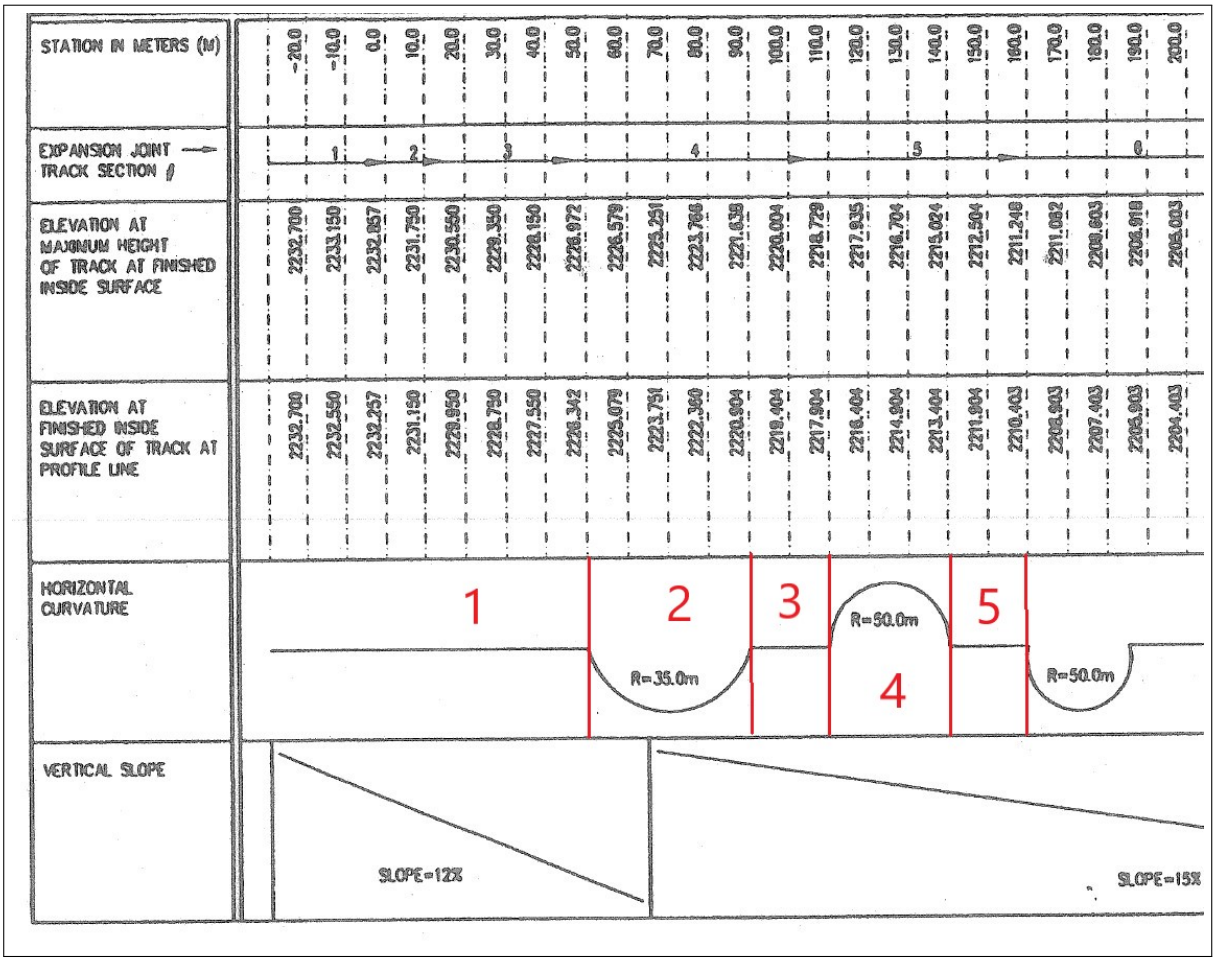


Figure 2.5: Numerical Source - Using Horizontal Curvature to divide first 160 m of Track Profile Line into 5 segments

As in roads and railway track lines, the transition for straight segment (radius of curvature = ∞) to a segment with a finite radius of curvature or vice-versa is not instantaneous but gradual and can be described by 'clothoids' [11]. A clothoid is a curve whose radius of curvature increases or decreases linearly along the length of the curve. This information is missing from Numerical Source and leads to a Track Profile Line that varies greatly from the one obtained from Graph Source as shown in Fig. 2.6.

Corrections were made to the line obtained using Numerical Source by trial and error, varying the ratio of the length of a curved segment that is comprised of clothoids, starting from curve - 1 and then continuing forward. Furthermore, each curved segment is assumed to have a clothoid of same length at the start and at the end. For example, if the clothoid length to segment length ratio is 0.25. The entry clothoid is 0.125 of curve length and so too is the exit clothoid. The clothoid at the start of a curved segment, the radius of curvature decreases linearly, while it increases for the clothoid at the end of a segment. This proved suitable for all curves apart from curve - 6 which appears to not have a clothoid at the end of the segment. Figures 2.6 and 2.7 compare the two profile lines before and after addition of clothoids to the line obtained from Numerical Source.

Now that the Track Profile Line in XY plane has been obtained the next step is to determine the line in 3D space. From the Numerical Source, 'Elevation At Finished Inside Surface of Track At Profile Line' contains elevation data measured every 10 metres along the Track Profile Line. When a value of elevation is required for a point, whose distance is not a multiple of 10, it is obtained by linear interpolation. The Track Profile Line in 3D space is shown in Fig. 2.8.

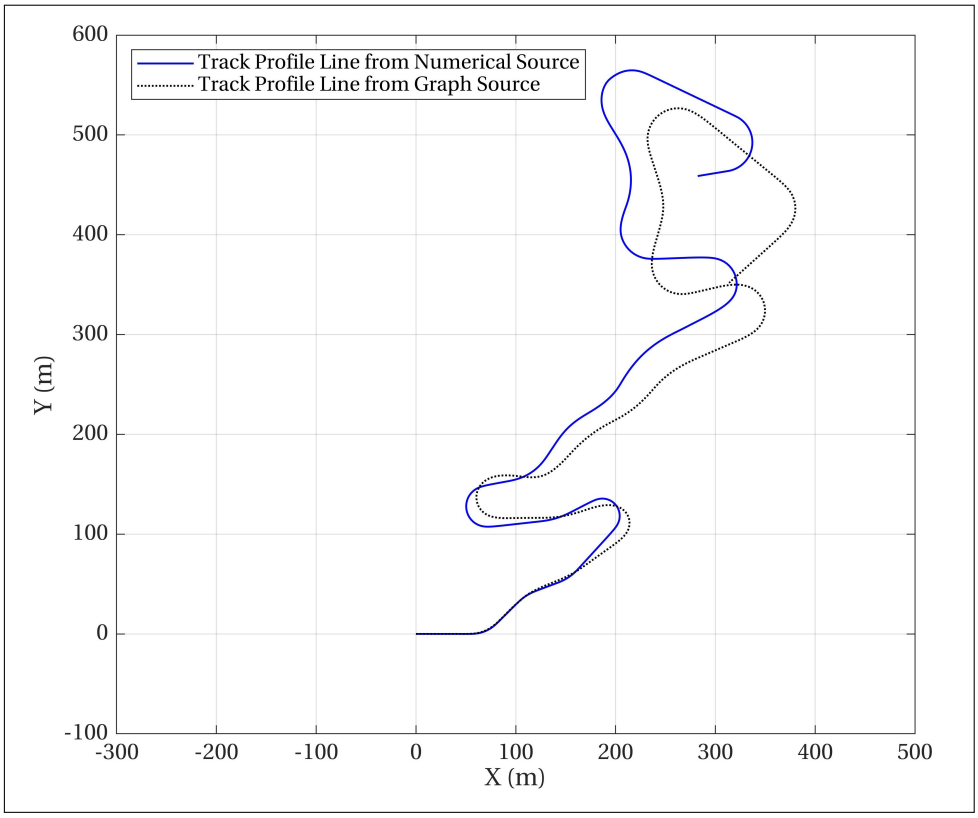


Figure 2.6: Comparison of Track Profile Lines obtained from the two sources, Numerical Source without clothoids

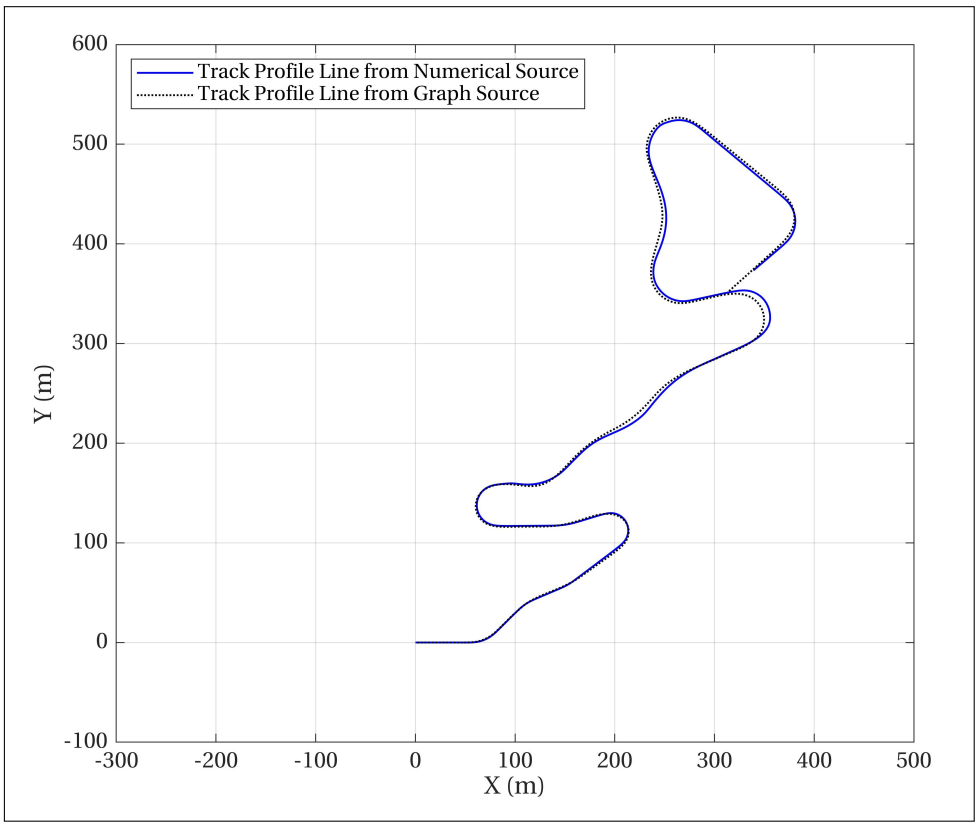


Figure 2.7: Comparison of Track Profile Lines after clothoids are added to the line from Numerical Source

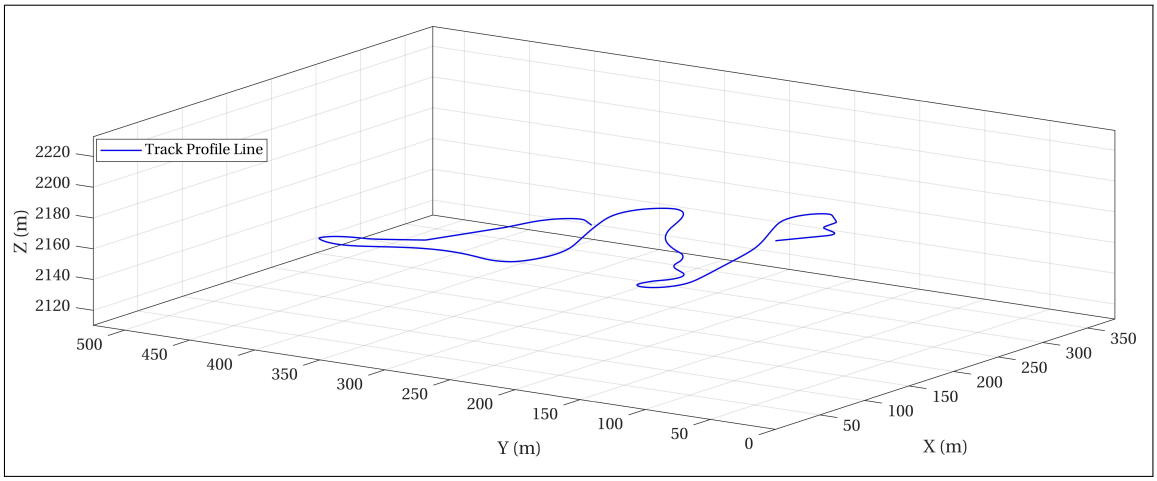


Figure 2.8: Track Profile Line after addition of elevation as z-axis co-ordinates.

Discretization and Parametrization
To be able to assemble the transverse cross-sections in the next step, we need points along the profile line. For straight segments we take a point every 0.5 m. For a curved segment, the number of points is determined by the number of frames obtained from the video for that segment. For each point, a direction angle θ_d and a slope angle θ_s are computed.

The direction angle θ_d is determined from the slope of the tangent to the profile line in XY plane as shown in Fig. 2.9. The slope angle θ_s is determined as follows

$$\theta_s = \arctan \frac{(z_{i+1} - z_i)}{s_i}, \tag{2.1}$$

where s_i is the arc length between point i and $(i+1)$ measured along the Track Profile Line. These two angles along with (x_i, y_i, z_i) coordinates of a point determine the rotation and translation of transverse cross-sections.

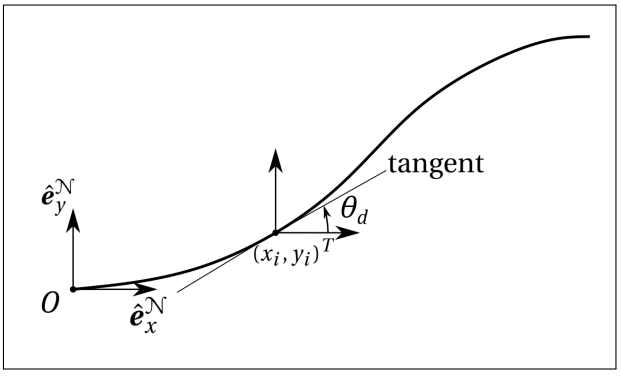


Figure 2.9: Determining the direction angle at point $(x_i, y_i)^T$ of the Track Profile Line using the tangent to the line at that point. $\boldsymbol{O}$ is the origin of the coordinate system associated with frame $\mathcal{N}$

2.1.2. Assembly of Transverse Cross-sections

Approximation of Transverse Cross-Sections
The approximation of the transverse cross-sections is different for straight segments and curved segments of the track, while the number of points that describe each cross-section is always 80.

Straight Segment
The transverse cross-sections of straight segments is assumed to be the same for all such segments of the

track. The dimensions of this “standard cross-section" used here is taken from [11], it describes the dimensions of transverse cross-section of the track at Whistler, Canada and is shown in Fig. 2.10.

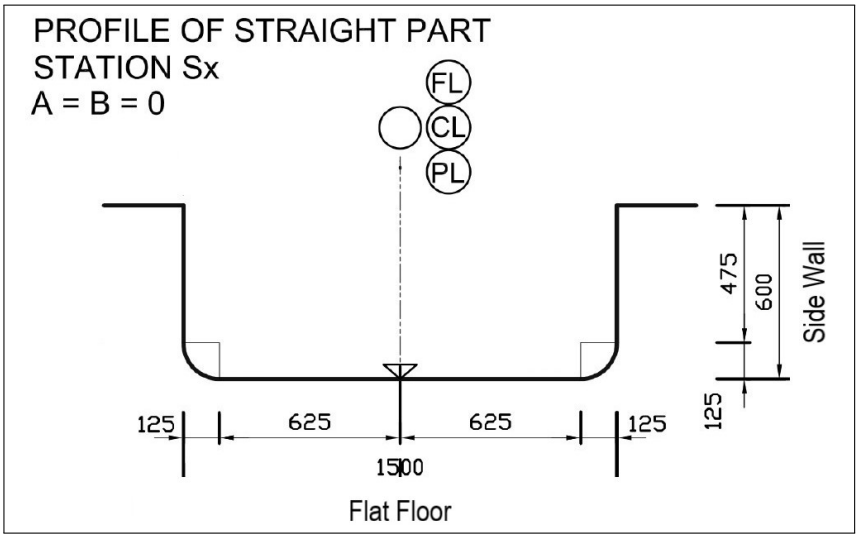


Figure 2.10: Standard cross-section of track surface in all straight segments. PL stands for Profile Line. CL for Centre Line. FL for Foundation Line. This image is from [11]

Curved Segment

For curved sections, the transverse cross-sections are approximated using video footage of a descent down the track. Frames (still images) during traversal of a curved segment are extracted from the video footage. For each frame, points are marked using ImageJ software [12]. An example is shown in Figure 2.11. The first 4 points comprise the side wall. These four points are used to rotate and scale the cross section. The distance between points 1 and 4 is assumed to be equal to the height of the side wall in the cross-sections of straight segments (0.475 m, Figure 2.10) and is used to scale the cross-section. The angle between the line formed by joining point 1 to point 4 and y axis of the image is used to rotate the cross-section such that the line is parallel to y axis of the image as shown in Fig. 2.11. Between point 4 and 5 is assumed to be a quarter circle (of radius 0.125 m, Fig. 2.10) between side wall and flat floor. The points 5 to 8 represent approximately half of the flat floor (0.625 m, Fig. 2.10) of the cross-section. Finally a Bezier curve [13] is interpolated between points 8 and 20. More details about Bezier curves can be found in Appendix B. The resulting cross-section is shown in Figure 2.12.

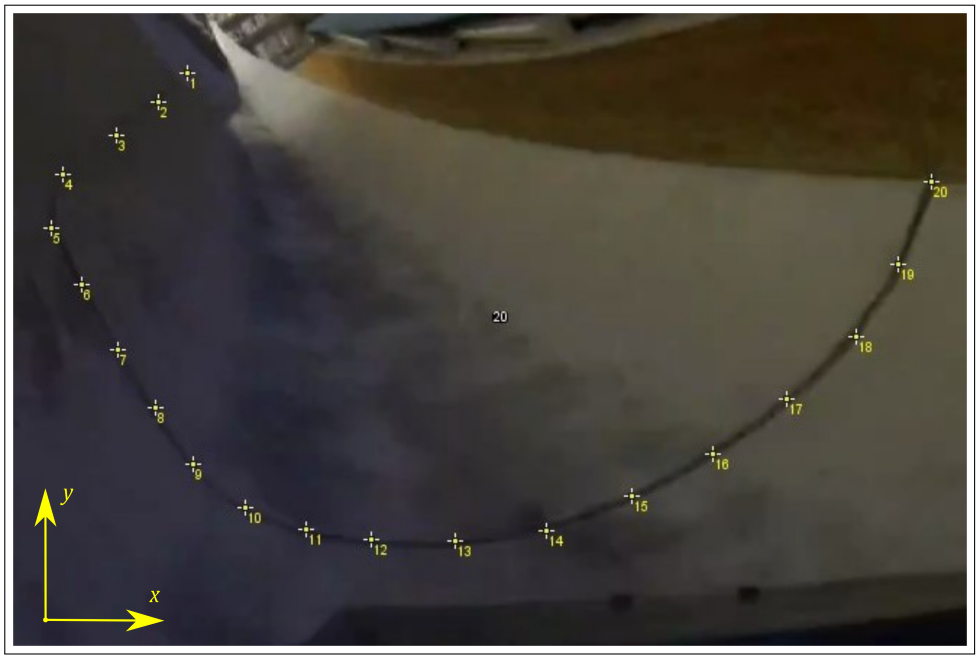


Figure 2.11: Example of using ImageJ to approximate the transverse cross section of curved segments of the track. The x and y axes form a local co-ordinate system to represent locations of pixels in the image

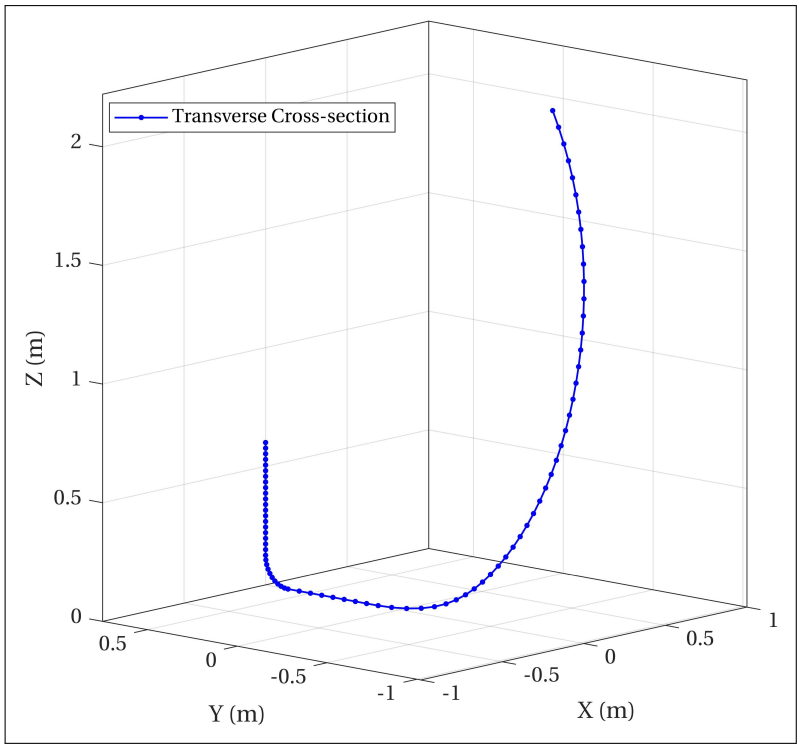
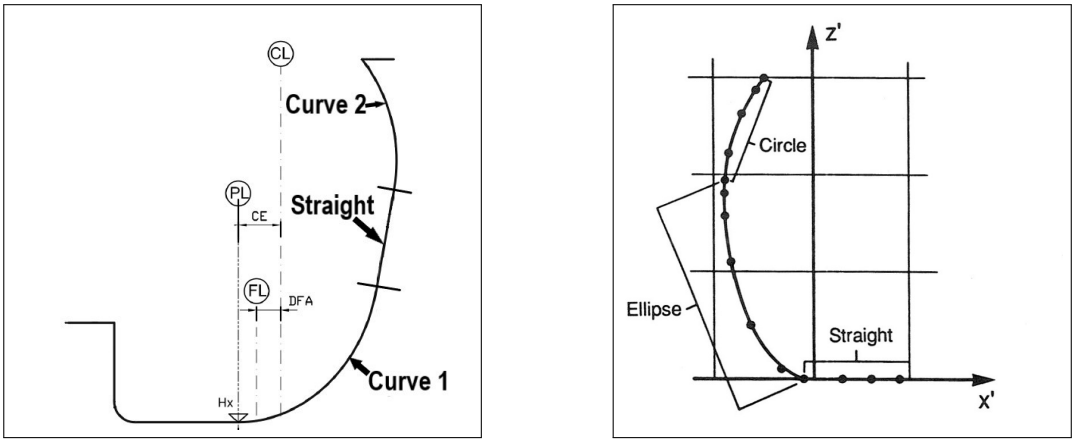


Figure 2.12: Transverse cross section of 80 points obtained by interpolating between the 20 points of Figure 2.11

Choice of Bezier Curves

Figures 2.13a and 2.13b show examples of the actual shape of a transverse cross-section. Such curves with multiple types of conic sections would be better represented by B-Splines [6, 14], rather than Bezier curves (which are a special case of B-Splines). But in the frames obtained from the video, we could neither determine the transition between conic sections nor the nature of these sections with reasonable confidence. Though in most cases it was possible to determine when the cross-section transitioned from the straight line of the floor

to the curve, i.e. the start of the curve. The end-point interpolation property of Bezier curves [**L1.1**] is useful here. Hence the decision to approximate a single smooth Bezier curve from the start of the curve (point 9 in Fig. 2.11) to the end of the curve (point 20 in Fig. 2.11).



(a) From [11]

(b) From [14]

Figure 2.13: Examples of the actual shape of transverse cross-sections

Assembly of Transverse Cross-sections
The next step is to assemble these cross-sections along the Track Profile Line. Each cross-section is first rotated by the direction angle θ_d about Z axis and then by the slope angle θ_s about Y axis. After rotation, the cross-section is translated to its position along the Track Profile Line. Figure 2.14 shows transverse cross-sections aligned along the Track Profile Line for curve - 4 of the track. During a descent the athlete and sled experience increases and decreases in velocity during traversal of a curved segments. This is not taken into consideration when assembling the transverse cross-sections of these segments. Instead, cross-sections are placed along the Track Profile Line such that they are equally distant to each other.

2.1.3. Interpolation Between Cross-sections

The minimum distance between individual cross-sections is 0.5 m in the straight segments. This distance is even larger in curved segments. If one were to carry out triangulation now, it would lead to triangles with a bad aspect ratio, that in turn adversely effect the simulation [6]. To reduce this distance we obtain more cross-sections using interpolation. Here again, Bezier curve interpolation is used to obtain additional cross-sections along the Track Profile Line. The interpolation is carried out at the segment level, that is the we start by interpolating between the cross-sections of segment 1 only and then segment 2 only and so on.

Let the points obtained after assembling all cross sections as shown in Fig. 2.14 be represented by $\boldsymbol{p}_{k,i}$ where k identifies which cross section the point belongs to (along track distance) and i identifies the point within the cross section (across track). Since each cross-section consists of 80 points $i = 1, 2, 3, ..., 80$. Here, we interpolate between the points $\boldsymbol{p}_{k,i}$ for a given i, so for each segment we are interpolating 80 curves. For a straight segment, the parameter u of the Bezier curves is evaluated such that we obtain a cross-section every 0.1 m, whereas each curved segment is made to comprise of 500 cross-sections.

Choice of Bezier Curves
While interpolating between cross-sections of a track segment, from the Numerical Source data we see that a curved segment is preceded and succeeded by a straight segment. This is true for the track at Park City, we are not aware if it is true for all other tracks. Thus we know the start and end of the curves with reasonable confidence. The end-point interpolation property of Bezier curves (**L1.1**) is handy once again.

For a physics simulation one aims to attain surfaces with C^2 continuity. Due to the infinite differentiability of B-splines in the interior span between two knots (property 2 of Bezier curves, **L1.2**), when interpolating between cross-sections of a segment, continuity only needs to be considered at the end points. To achieve

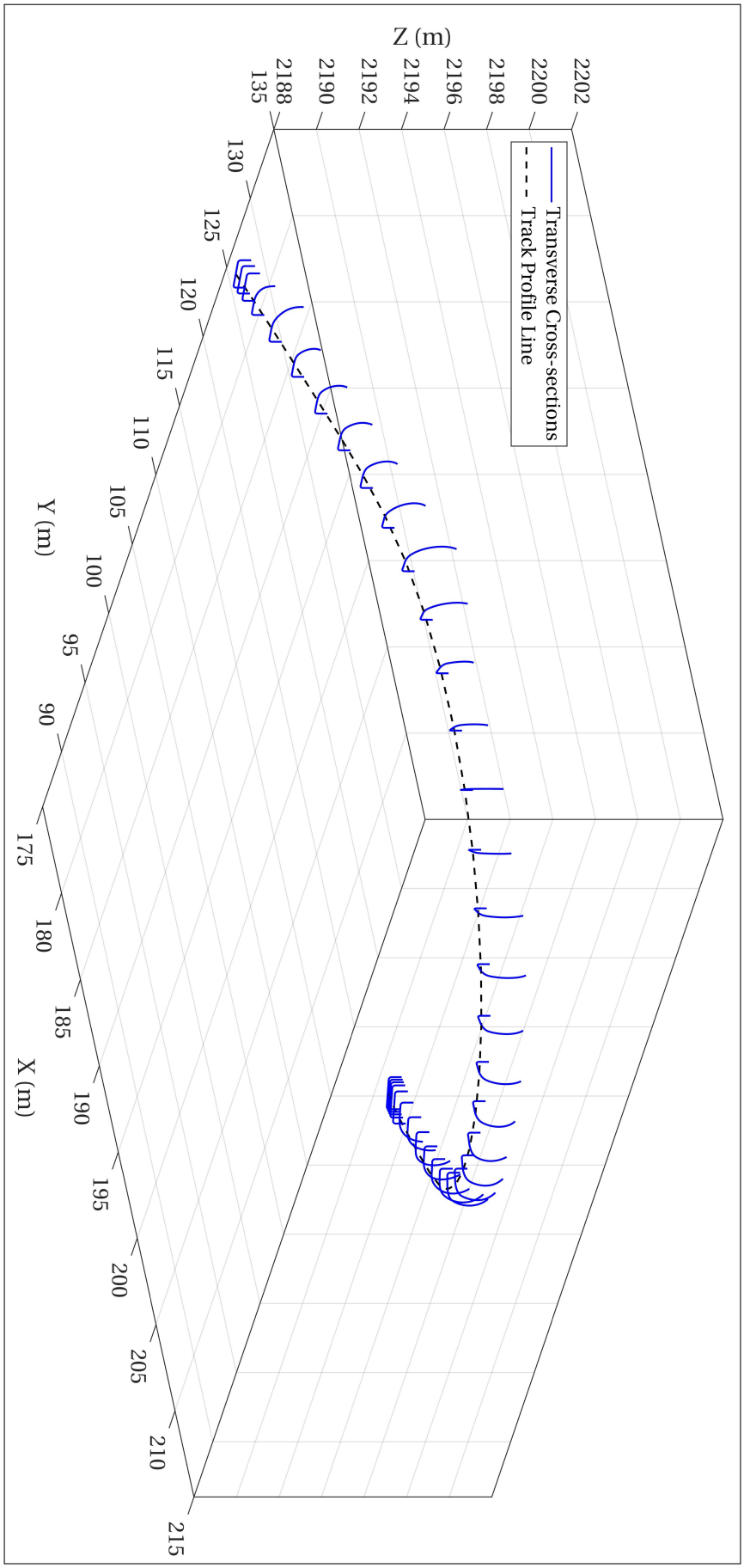


Figure 2.14: Assembly of Transverse Cross-sections of curve 4 of track at Park City, Utah. The blue lines join points $\boldsymbol{p}_{k,i}$ with constant k.

some level of continuity at the end points, the cross-sections that comprise a segment will be slightly altered. If one were to consider the last three cross sections of a straight section as part of the succeeding curved segment then the points $\boldsymbol{p}_{k-2,i}, \boldsymbol{p}_{k-1,i}, \boldsymbol{p}_{k,i}, \boldsymbol{p}_{k+1,i}, \boldsymbol{p}_{k+2,i}$, where k is the cross-section at the altered end of the straight segment and altered start of the curved segment. All these points will lie in straight line. Since the cross-sections of a straight segment are equally distant from each other (0.5 m, this distance is before interpolation between cross-sections), the distances between these five points will also be equal. Subsequently Eqn. (B.12) will be satisfied resulting in G^2 between segments. The degree of the two Bezier curves will be different as the number of control points are usually different in our case, but since one of them is a straight line, degree elevation or reduction can be easily done to result in C^2 continuity at the junctions.

The use of video footage to approximate the individual transverse cross-sections, leads to noticeable differences in shape and scale of the cross-sections obtained. This can be clearly seen in Fig. 2.14. The global influence of a control point on the entire curve (**L1.3**) becomes useful. Properties 1 to 3 of Bezier curves (**L1.1** to **L1.3**) combine to result in one smooth curve for a given i when interpolating between cross-sections of a given segment. The cross-sections obtained after this round of interpolation are shown in Fig. 2.15. Above all the most convenient aspect of the above method is that the points determined from video frames themselves can be used as control points for the Bezier curves. Eliminating the need for computing them separately.

2.1.4. Triangulation

After interpolation between cross-sections, we obtain a distribution of points in 3D space. To obtain a surface, a triangular mesh is generated which approximates a planar surface defined by these points. Triangles are commonly used in computer graphics and physics simulations to represent plane surfaces as the minimum number of points needed to define a plane is 3. Each cross section is defined by 80 points and consequently, 158 triangles are obtained between two cross-sections. The final mesh contains approximately 1.7 million triangles and for each triangle, the centroid, surface unit normal and (two) unit tangential vectors are computed. Figures 2.16 and 2.17 show the 3D point cloud and the triangular mesh obtained from it. Along with ImageJ, MATLAB [15] is used throughout the track reconstruction process.

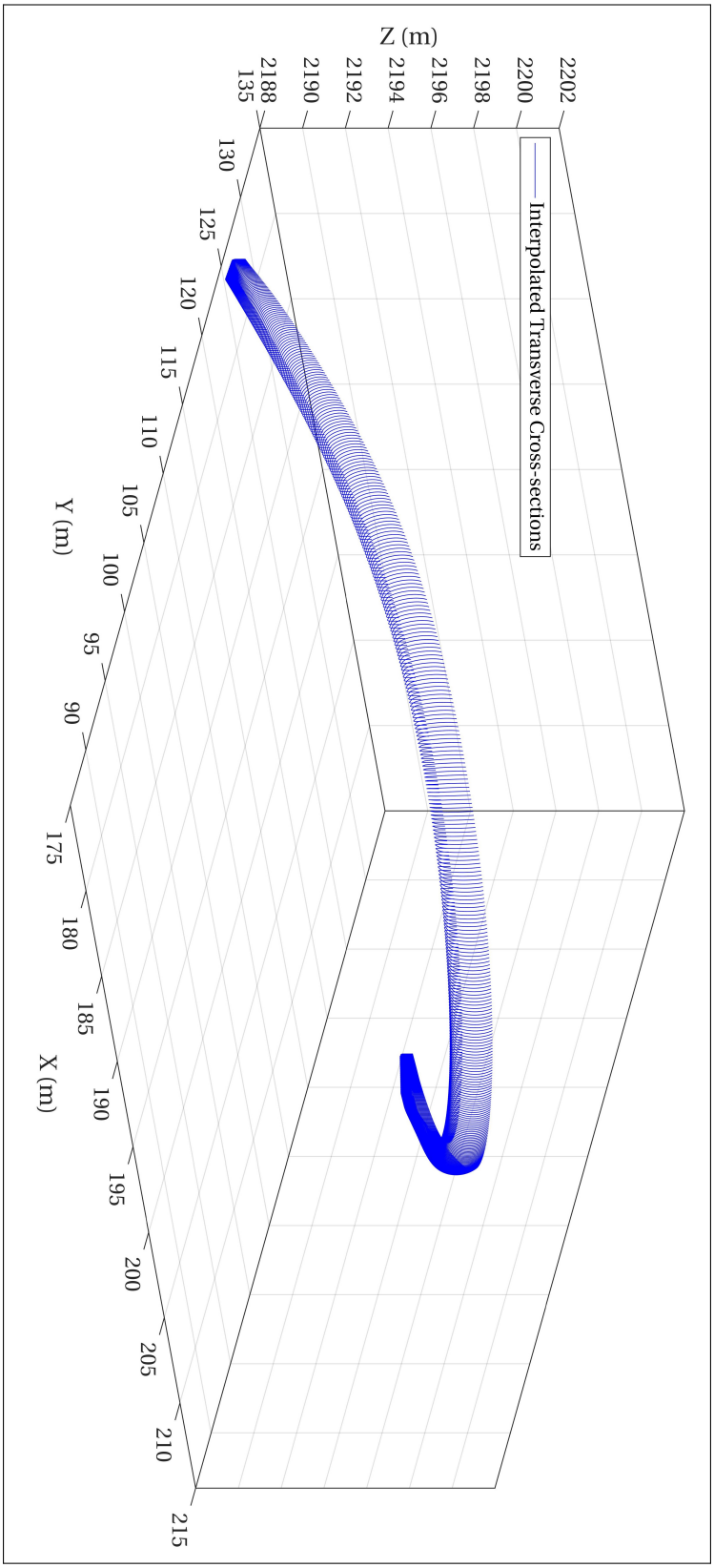


Figure 2.15: The result of interpolating between the cross sections for curve 4. The lines join points $\boldsymbol{p}_{k,i}$ with constant k.

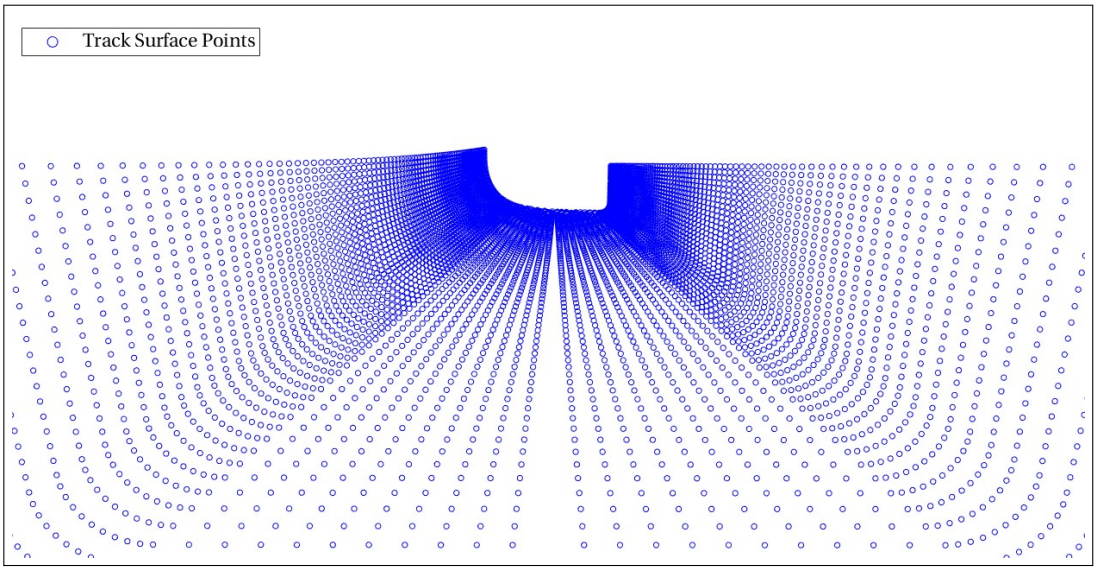


Figure 2.16: Distribution of points obtained after interpolating between transverse cross-sections

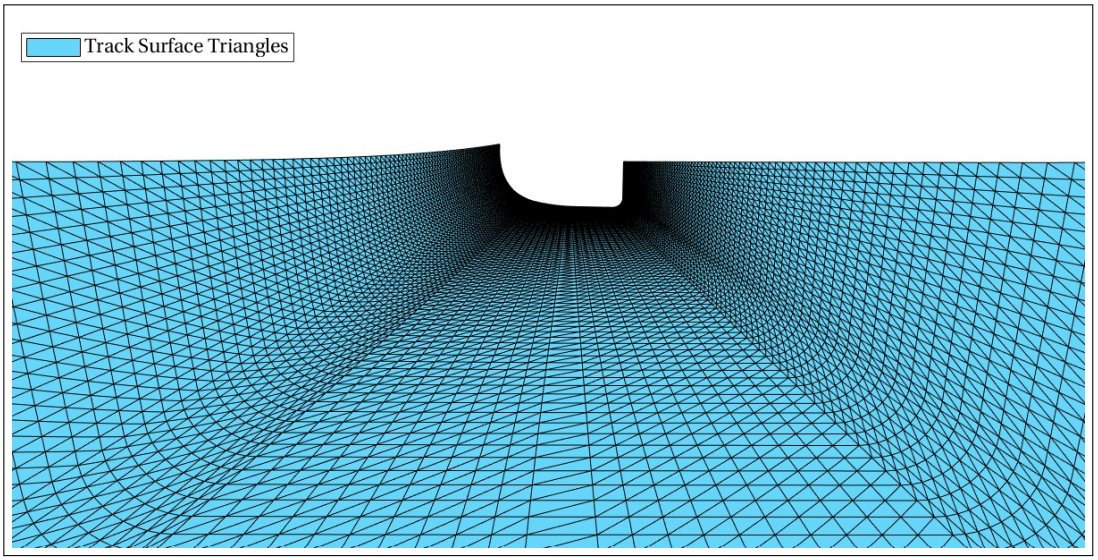


Figure 2.17: Triangular mesh obtained from the points in Fig 2.16 above

3

Dynamic Model

This chapter describes our model of athlete-sled body and more importantly the contact model between the sled and track surface as it descends down the track. For this simulator, we consider the motion of the sled once the athlete has loaded on to the sled at the end of push-start phase. Later on, the time-stepping scheme for the equations of motion is explained.

3.1. Sled Model

We assume the athlete and sled to be a single body, S. A frame $\mathcal{B}$, is attached to the centre of mass of this body. This body fixed frame is represented by a right-handed orthogonal coordinate system $X'Y'Z'$ with X' as the longitudinal axis (forward/backward directions), Y' as the lateral axis (left/right directions) and Z' as the vertical axis. The basis vectors of this frame are $\hat{\boldsymbol{e}}_x^{\mathcal{B}}, \hat{\boldsymbol{e}}_y^{\mathcal{B}}, \hat{\boldsymbol{e}}_z^{\mathcal{B}}$. The combined body has mass m and mass moments of inertia ${}^{\mathcal{B}}I_{xx}$, ${}^{\mathcal{B}}I_{yy}$, ${}^{\mathcal{B}}I_{zz}$. For the mass moments of inertia of combined athlete-sled body we consider a cuboid body, having dimensions of the athlete with height h, along X'. The width w, measured across frontal plane is along Y' and finally d, measured across sagittal plane is along Z'.

Each of the two runners is represented by a line of length $2a$. The runners are parallel to X' axis at all times. The mid-point of each runner is fixed with respect to the centre of mass . These mid-points are a distance c below the centre of mass and a distance b on either side of the centre of mass as shown in Fig. 3.1. The runners are assumed to make contact with the track surface at the end points of the line. Thus there are 2 contact points for each runner. At their mid-points, each runner can rotate about an axis parallel to Y' axis. These two rotations are constrained such that they are equal and opposite to each other. α is the relative angle between the two runners measured about an axis parallel to Y' axis. A mass moment of inertia I_α is associated with α. While the dimensions b and c are very similar to the height and width of the sled. The length of the sled can range from 0.8 m to 1.2 m. We assume the centre of mass to be located at the centre of a 1 m long sled. With this assumption the athlete head would be displaced from S by approximately ${}^{\mathcal{B}}(0.5, 0, 0)^T$. Henceforth, when we say sled, we mean the combined athlete-sled body.

3.2. Forces

The Inertial Reference Frame $\mathcal{N}$ used for the dynamic model is the same as that used for the track reconstruction, represented by a right handed orthogonal coordinate system XYZ with gravity acting along $-\hat{\boldsymbol{e}}_z^{\mathcal{N}}$. We categorise the forces acting on the sled into two types:

1. Contact forces:
    - Normal (reaction) forces
    - Friction forces
2. Non-Contact forces:
    - Gravity

- Aerodynamic forces

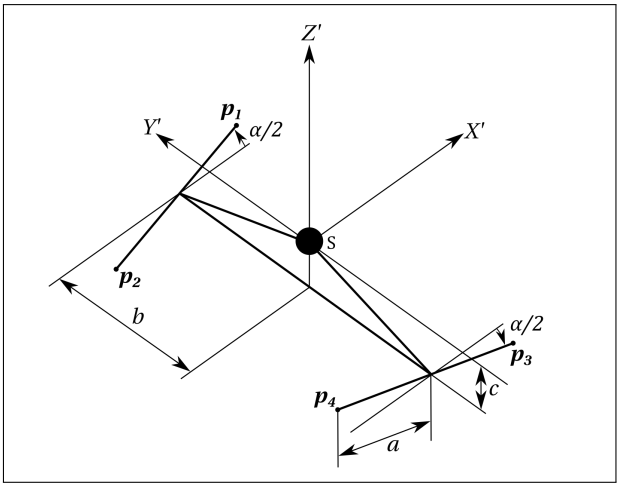


Figure 3.1: The figure shows the sled body S with axes X', Y', Z' of frame $\mathcal{B}$. The dimensions a, b, c and angle α define the position of four contact points $\boldsymbol{p}_k$ w.r.t. S.

3.2.1. Contact Forces

We will only consider the contact forces that arise due to interaction of sled runners with the track surface. Forces that arise due to contact of other elements of the sled with the track surface are neglected.

As mentioned in Section 3.1, the sled has four contact points and a contact is considered to be “closed" if the the contact point lies on the track surface. The model assumes that all the four contacts are closed for the entire duration of a descent, that is, bilateral constraints. The track is represented by a triangular mesh as described in Chapter 2. During each time step of the simulation, the nearest triangle centroid to each contact point is determined. Let $\boldsymbol{p}_k$ be the position of a contact point and $\boldsymbol{c}_k$, $\hat{\boldsymbol{n}}_k$ be the position of the centroid and unit normal vector of the nearest triangle (to point $\boldsymbol{p}_k$) respectively. These three vectors are measured in Inertial frame $\mathcal{N}$ coordinate system. Then the displacement $\boldsymbol{r}_k$, between the contact point and triangle centroid is

$$\boldsymbol{r}_k = \boldsymbol{p}_k - \boldsymbol{c}_k, \qquad k = 1,2,3,4. \tag{3.1}$$

The dot product of $\boldsymbol{r}_k$ with $\hat{\boldsymbol{n}}_k$ gives the distance of contact point $\boldsymbol{p}_k$ from the surface. For the contact to be closed, this distance must be zero. Thus the k constraint equations, one for each contact point are given by

$$C_k = \boldsymbol{r}_k \cdot \hat{\boldsymbol{n}}_k = 0. \tag{3.2}$$

A closed contact experiences a normal (reaction) force and friction forces. Figure 3.2 shows a single contact point and the vectors associated with the constraint equation.

Normal (Reaction) Forces

The magnitude of a Lagrangian multiplier λ_k, represents the magnitude of normal reaction force that acts on a closed contact. The direction of these normal forces is the same or opposite to that of the respective normal vector, $\hat{\boldsymbol{n}}_k$ when the contact point is above or below the surface respectively. Thus the net normal force is given by

$$\boldsymbol{f}_N = \sum_{k=1}^{4} \operatorname{sgn}(\boldsymbol{r}_k \cdot \hat{\boldsymbol{n}}_k)\, |\lambda_k|\, \hat{\boldsymbol{n}}_k. \tag{3.3}$$

In Figure 3.2, we show a single contact point and the vectors related to the normal reaction force. Since these normal reaction forces are implicitly taken care of through the constraint equations $\boldsymbol{C}_k$, we omit them and the resulting moments in the remainder of this chapter.

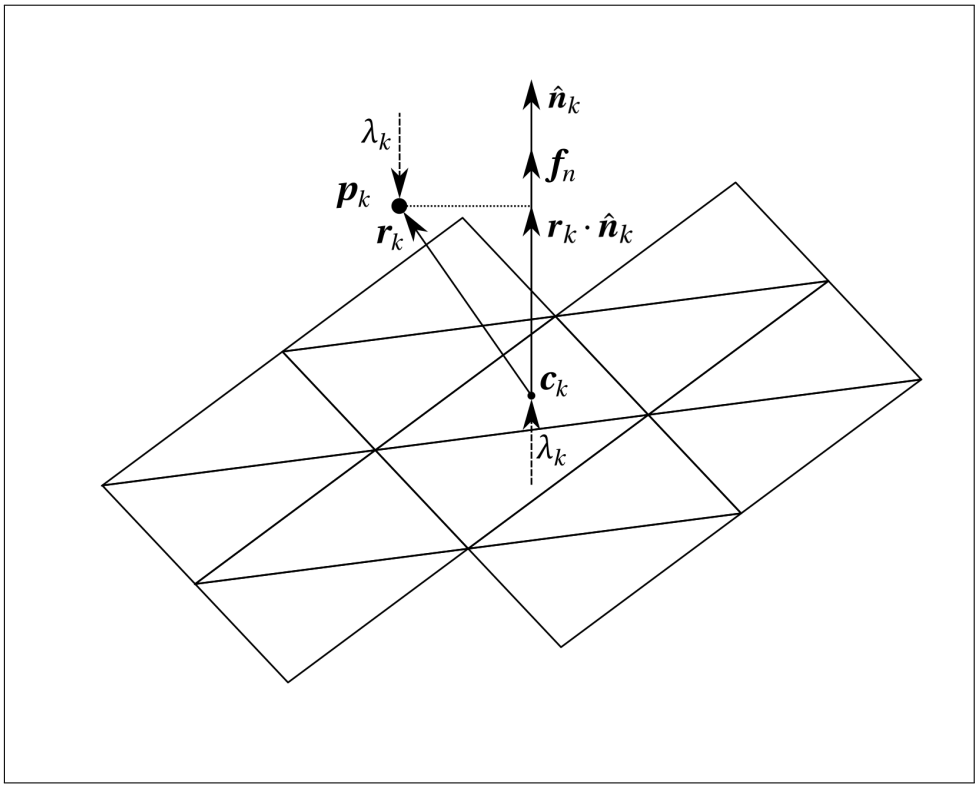


Figure 3.2: This figure shows a contact point $\boldsymbol{p}_k$, the centroid $\boldsymbol{c}_k$ and normal vector $\hat{\boldsymbol{n}}_k$ of the nearest triangle. Enforcing the constraint $C_k = \boldsymbol{r}_k \cdot \hat{\boldsymbol{n}}_k = 0$, leads to Lagrangian multiplier λ_k whose action is represented by the dashed lines. Finally we show the normal reaction force $\boldsymbol{f}_n$ acting on $\boldsymbol{p}_k$.

Friction Forces

At each contact point $\boldsymbol{p}_k$, a friction force acts in the direction opposite to that of the relative velocity between contact point and track surface. As we have a bilateral constraint at each contact point and a static track surface, the relative velocity reduces to the velocity of contact point itself. Figure 3.3 shows the sled when viewed from above. Let S be moving with a linear velocity $\boldsymbol{v}_S$ and let ${}^{\mathcal{B}}\boldsymbol{\omega}$ be the angular velocity of S in frame $\mathcal{B}$. The velocity of a contact point $\boldsymbol{p}_k$ is given by

$$\boldsymbol{v}_k = \boldsymbol{v}_S + {}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}}({}^{\mathcal{B}}\boldsymbol{\omega} \times {}^{\mathcal{B}}\boldsymbol{p}_k), \tag{3.4}$$

where, rotation matrix ${}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}}$ rotates from $\mathcal{B}$ to $\mathcal{N}$ and ${}^{\mathcal{B}}\boldsymbol{p}_k$ is the position of contact point k with respect to S in frame $\mathcal{B}$. We do not consider the effect of α here hence ${}^{\mathcal{B}}\boldsymbol{p}_k$ is constant. This velocity can be resolved into $\boldsymbol{v}_{long}$ parallel to the X' axis and $\boldsymbol{v}_{lat}$ parallel to Y' axis as shown in Figure 3.3.

The simulation begins after the athlete has loaded onto the sled and and ideally ends when they cross the finish line, at no point during a competitive descent should the sled come to a stop. In other words, we do not encounter stick (static) state and are always in a state of slip (dynamic) friction in the longitudinal direction. This longitudinal friction force acting opposite to $\boldsymbol{v}_{long}$ is modelled using a simple Coulomb dry friction model and is given by

$$\boldsymbol{f}_C = -{}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}}\,\hat{\boldsymbol{e}}_x^{\mathcal{B}} \sum_{k=1}^{4} \mu_k\,|\lambda_k|, \tag{3.5}$$

where, μ_k is the coefficient of friction at $\boldsymbol{p}_k$.

To quantify the lateral friction forces observed during motion of the sled, we use a model applied to tires in vehicle dynamics. According to this model, the lateral force is the product of the normal contact force, the angle between the longitudinal axis of the tire and velocity of the tire. For our case, the angle δ_k is the angle between the velocity vector of the contact point $\boldsymbol{v}_k$ and $\boldsymbol{v}_{long}$ as shown in Fig. 3.3. The lateral friction force $\boldsymbol{f}_L$, acting opposite to $\boldsymbol{v}_{lat}$ is given by

$$\boldsymbol{f}_L = -{}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}}\,\hat{\boldsymbol{e}}_y^{\mathcal{B}} \sum_{k=1}^{4} \delta_k C_S \,|\lambda_k|\,, \tag{3.6}$$

where, the scalar C_S is the cornering stiffness parameter.

The friction forces acting at the contact points lead to moments about the centre of mass S which are

$${}^{\mathcal{B}}\boldsymbol{m}_C = -\sum_{k=1}^{4} {}^{\mathcal{B}}\boldsymbol{p}_k \times \mu_k \,|\lambda_k|\,\hat{\boldsymbol{e}}_x^{\mathcal{B}}, \tag{3.7}$$

$${}^{\mathcal{B}}\boldsymbol{m}_L = -\sum_{k=1}^{4} {}^{\mathcal{B}}\boldsymbol{p}_k \times \delta_k C_S \,|\lambda_k|\,\hat{\boldsymbol{e}}_y^{\mathcal{B}}. \tag{3.8}$$

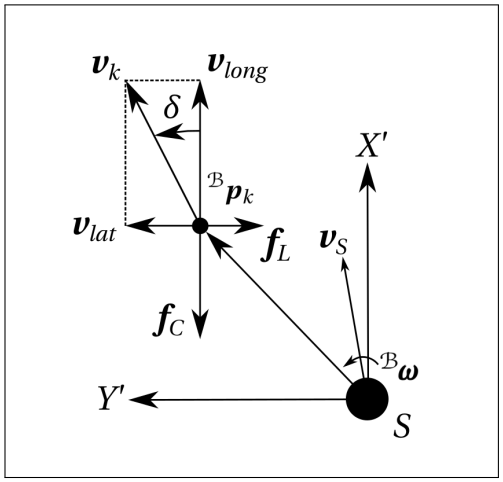


Figure 3.3: Figure shows the decomposition of velocity of contact point k $\boldsymbol{v}_k$, into two components and the friction forces that act opposite to the velocities.

3.2.2. Non-contact Forces

Gravity

The force due to gravity $\boldsymbol{f}_G$ is

$$\boldsymbol{f}_G = -mg\hat{\boldsymbol{e}}_z^{\mathcal{N}}, \tag{3.9}$$

where, m is the mass of sled, g is the acceleration due to gravity and is equal to 9.81 m/s^2.

Aerodynamic Forces

As the sled descends down the track it is subject to two aerodynamic forces. First, a resistive drag force that acts in the direction opposite to its velocity. Second, a lift force that acts normal to the velocity in the positive z-axis direction of the body-fixed frame $\mathcal{B}$. Here lift force is neglected and only the drag force $\boldsymbol{f}_D$ is considered and is

$$\boldsymbol{f}_D = -\frac{1}{2} C_D A_F \rho |\boldsymbol{v}_S| \boldsymbol{v}_S, \tag{3.10}$$

where, C_D is the coefficient of drag, A_F is the frontal area, ρ is the density of air and $\boldsymbol{v}_S$ is the linear velocity of S.

3.3. Athlete Steering Input

3.3.1. Shoulder and Knee Steering

One method by which a Skeleton athlete steers is by applying forces with their shoulders and knees on the sled to deform it. This deformation changes the contact location and geometry between the runners and the

ice surface leading to a change in friction force distribution and in turn to lateral forces and a moment. An athlete has the freedom to apply forces using a single shoulder or single knee or combinations of shoulder and knee, but due to the relative angle constraint between the two runners in our model the user effectively has only two choices. They are using right shoulder with left knee simultaneously or left shoulder with right knee.

This is realized through a generalized moment M_α acting on the relative angle between the two runners α. Consider the two runners and track surface shown in Fig. 3.4 and let a moment couple M_α act on the relative angle α between them. This generalized moment has the effect to reduce the Lagrangian multipliers λ_1 and λ_4, while increasing λ_2 and λ_3. Since the friction force $\boldsymbol{f}_C$ and lateral force $\boldsymbol{f}_L$ are functions of λ values, this changes the distribution of friction forces and the moments $\boldsymbol{m}_C$ and $\boldsymbol{m}_L$. The final result is an acceleration about the body Z' axis.

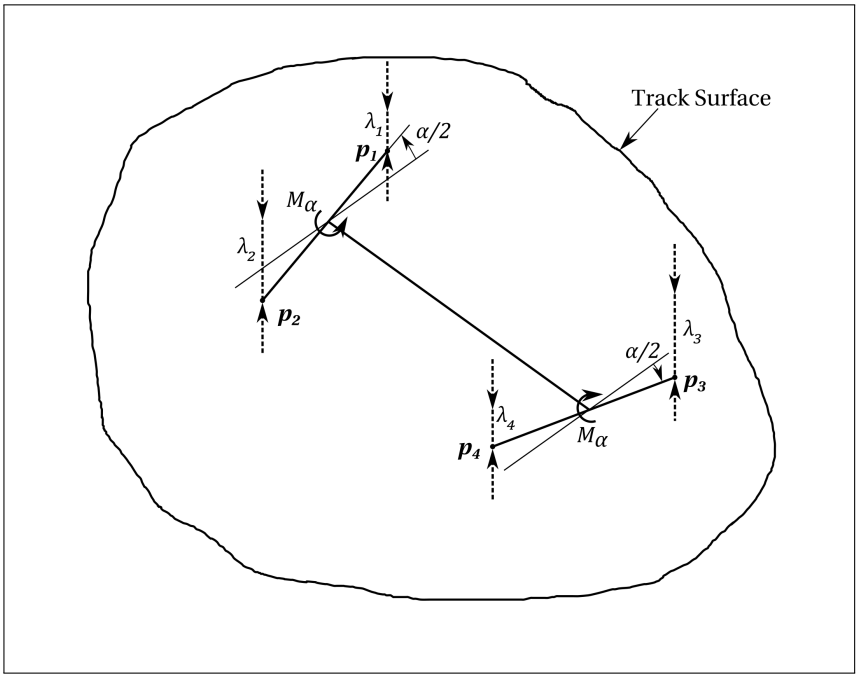


Figure 3.4: The effect of generalised moment M_α on Lagrangian multipliers $\boldsymbol{\lambda}$. A positive M_α has the effect of reducing the magnitudes of λ_1 and λ_4 while increasing λ_2 and λ_3. A negative M_α will increase λ_1 and λ_4 while reducing λ_2 and λ_3

3.3.2. Head Steering

A second method of steering involves tilting their heads to one side, changing the lateral distribution of aerodynamic drags forces and distribution of mass over the two runners to create small turning moments. This method of steering is realized using an external moment M_{HS} that acts on S about Z' axis.

3.3.3. Toe Steering

As toe steering is considered to be an emergency steering manoeuvre that considerably reduces velocity and used in extreme cases only, it is neglected. More details about steering in Skeleton can be found in Appendix A.

3.4. Equations of Motion

For a rigid body, the constrained equations of motion can be written in the form of Differential Algebraic Equations (Eqn. (12.11) of [16]) as

$$\begin{pmatrix} \boldsymbol{M}_{ij} & \boldsymbol{C}_{k,i} \\ \boldsymbol{C}_{k,j} & \boldsymbol{0}_{kk} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{x}}_j \\ \boldsymbol{\lambda}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_i \\ -\boldsymbol{C}_{k,jl}\dot{\boldsymbol{x}}_j\dot{\boldsymbol{x}}_l \end{pmatrix}, \tag{3.11}$$

where, $\boldsymbol{M}_{ij}$ is the mass matrix, $\boldsymbol{C}_{k,j}$ is the Jacobian of the k constraints $\boldsymbol{C}_k$ with respect to the j generalized coordinates $\boldsymbol{x}_j$ and $\boldsymbol{C}_{k,i} = \boldsymbol{C}_{k,j}^T$. The vector $\ddot{\boldsymbol{x}}_j$, are accelerations of generalized coordinates $\boldsymbol{x}_j$. $\boldsymbol{\lambda}_k$ are the Lagrangian multipliers for the constraints $\boldsymbol{C}_k$, whose physical meaning depends on the constraints. The vector $\boldsymbol{f}_i$ contains external forces and moments acting on the generalised coordinates. Finally $-\boldsymbol{C}_{k,jl}\dot{\boldsymbol{x}}_j\dot{\boldsymbol{x}}_l$ contains the convective terms of the constraint equations.

The generalized co-ordinates $\boldsymbol{x}_j$ are

$$\boldsymbol{x}_j = \left(\boldsymbol{p}_S,\ \boldsymbol{q}_S,\ \alpha\right)^T,$$

where, $\boldsymbol{p}_S = (x,\ y,\ z)^T$ represents the position of the centre of mass of the body S. The orientation of the body-fixed frame $\mathcal{B}$ w.r.t. $\mathcal{N}$ is parametrized using Euler Parameters, represented by the unit quaternion $\boldsymbol{q}_S = (q_0, \boldsymbol{q})^T$. With the use of Euler parameters, the mass matrix $\boldsymbol{M}_{ij}$ is no longer constant. α is the relative angle between the two runners. Later on, we also use a Yaw-Pitch-Roll convention for describing the orientation of the sled. In that context, X is the roll axis, Y is the pitch axis and Z is the yaw axis.

The vector $\boldsymbol{f}_i$ contains the contact forces that are functions of the unknown Lagrangian multipliers $\boldsymbol{\lambda}_k$ and can be moved the L.H.S. of Eqn. 3.11 using the matrix $\boldsymbol{W}$ defined as

$$\boldsymbol{W} = \begin{pmatrix} \frac{\partial \boldsymbol{f}_C}{\partial \boldsymbol{\lambda}_k} + \frac{\partial \boldsymbol{f}_L}{\partial \boldsymbol{\lambda}_k} \\ \\ \frac{\partial {}^{\mathcal{B}}\boldsymbol{m}_C}{\partial \boldsymbol{\lambda}_k} + \frac{\partial {}^{\mathcal{B}}\boldsymbol{m}_L}{\partial \boldsymbol{\lambda}_k} \\ \\ \boldsymbol{0}_k \end{pmatrix}. \tag{3.12}$$

$\boldsymbol{f}_i$ can now be written as

$$\boldsymbol{f}_i = \begin{pmatrix} \boldsymbol{f}_G + \boldsymbol{f}_D \\ \boldsymbol{e} \\ M_\alpha \end{pmatrix} + \boldsymbol{W}\boldsymbol{\lambda}_k, \tag{3.13}$$

where $\boldsymbol{e}$ arises from the use of Euler parameters and is,

$$\boldsymbol{e} = 2\boldsymbol{Q}\begin{pmatrix} 0 \\ {}^{\mathcal{B}}\boldsymbol{M}_C \end{pmatrix} + 8\dot{\boldsymbol{Q}}\begin{pmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{0} & {}^{\mathcal{B}}\boldsymbol{I}_C \end{pmatrix}\dot{\boldsymbol{Q}}^T\begin{pmatrix} q_0 \\ \boldsymbol{q} \end{pmatrix}. \tag{3.14}$$

The reader is referred to Chapter 19 and Appendix D of [16] for further reading on the terms that comprise $\boldsymbol{e}$.

Using (3.12) and (3.13), Equations (3.11) now become

$$\begin{pmatrix} \boldsymbol{M}_{ij} & \boldsymbol{C}_{k,i} - \boldsymbol{W} \\ \boldsymbol{C}_{k,j} & \boldsymbol{0}_{kk} \end{pmatrix}\begin{pmatrix} \ddot{\boldsymbol{x}}_j \\ \boldsymbol{\lambda}_k \end{pmatrix} = \begin{pmatrix} \underline{\boldsymbol{f}}_i \\ -\boldsymbol{C}_{k,jl}\dot{\boldsymbol{x}}_j\dot{\boldsymbol{x}}_l \end{pmatrix}, \tag{3.15}$$

where,

$$\underline{\boldsymbol{f}}_i = \begin{pmatrix} \boldsymbol{f}_G + \boldsymbol{f}_D \\ \boldsymbol{e} \\ M_\alpha \end{pmatrix}.$$

In succeeding sections, we will no longer use the Einstein Summation convention for subscripts and the above system of equations are written as

$$\begin{pmatrix} \boldsymbol{M} & \boldsymbol{C}_{,\boldsymbol{x}}^T - \boldsymbol{W} \\ \boldsymbol{C}_{,\boldsymbol{x}} & \boldsymbol{0} \end{pmatrix}\begin{pmatrix} \ddot{\boldsymbol{x}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ -\boldsymbol{C}_{,\boldsymbol{x}\boldsymbol{x}}\dot{\boldsymbol{x}}\dot{\boldsymbol{x}} \end{pmatrix}. \tag{3.16}$$

3.5. Numerical Integration

For the numerical integration of accelerations obtained from (3.16), we use an implicit time-stepping scheme based on the Generalised-α method [17]. It was first developed for use in structural dynamics, it allows for

direct integration of second order differential equations and numerical dissipation of spurious high frequencies that result from finite element discretization methods. Our implementation of the α method has two parts. First, if the time at start of a time step is t_n and end of the step is t_{n+1} with $t_{n+1} = t_n + \Delta t$, the positions and velocities at t_{n+1} are defined using Newmark-β expansions (exactly as Eqn. (4) and (5) in [17]) shown below

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \dot{\boldsymbol{x}}_n + \Delta t^2 \left(\left(\frac{1}{2} - \beta \right) \ddot{\boldsymbol{x}}_n + \beta \ddot{\boldsymbol{x}}_{n+1} \right), \tag{3.17}$$

$$\dot{\boldsymbol{x}}_{n+1} = \dot{\boldsymbol{x}}_n + \Delta t \left((1-\gamma) \ddot{\boldsymbol{x}}_n + \gamma \ddot{\boldsymbol{x}}_{n+1} \right). \tag{3.18}$$

Second, we apply (6),(10-13) of [17] to (3.16). One way to think of this expanded system of equations is as weighted averages between quantities at t_n and t_{n+1}. Accelerations are weighted by the parameter α_m while remaining quantities are weighted by the parameter α_f as shown below

$$\begin{aligned} &\begin{pmatrix} (1-\alpha_m)\boldsymbol{M}_{n+1} & (1-\alpha_f)(\boldsymbol{C}_{,\boldsymbol{x}}^T - \boldsymbol{W})_{n+1} \\ (1-\alpha_m)(\boldsymbol{C}_{,\boldsymbol{x}})_{n+1} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{x}}_{n+1} \\ \boldsymbol{\lambda}_{n+1} \end{pmatrix} + \begin{pmatrix} \alpha_m \boldsymbol{M}_n & \alpha_f (\boldsymbol{C}_{,\boldsymbol{x}}^T - \boldsymbol{W})_n \\ \alpha_m (\boldsymbol{C}_{,\boldsymbol{x}})_n & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{x}}_n \\ \boldsymbol{\lambda}_n \end{pmatrix} - \\ &(1-\alpha_f) \begin{pmatrix} \boldsymbol{f}_{n+1} \\ -(\boldsymbol{C}_{,\boldsymbol{xx}} \dot{\boldsymbol{x}} \dot{\boldsymbol{x}})_{n+1} \end{pmatrix} + \alpha_f \begin{pmatrix} \boldsymbol{f}_n \\ -(\boldsymbol{C}_{,\boldsymbol{xx}} \dot{\boldsymbol{x}} \dot{\boldsymbol{x}})_n \end{pmatrix} = \boldsymbol{0}. \end{aligned} \tag{3.19}$$

If $\boldsymbol{x}_n$, $\dot{\boldsymbol{x}}_n$, $\ddot{\boldsymbol{x}}_n$ and $\boldsymbol{\lambda}_n$ are known, the only unknown quantities in the above 3 equations are $\ddot{\boldsymbol{x}}_{n+1}$ and $\boldsymbol{\lambda}_{n+1}$ and can be obtained using a Newton-Raphson type iteration. Below we present the time stepping algorithm that solves (3.17), (3.18), (3.19) and prevents drift-off from the constraint equations at position and velocity level.

3.5.1. The Time-stepping Algorithm

1. $\boldsymbol{x}_n$, $\dot{\boldsymbol{x}}_n$, $\ddot{\boldsymbol{x}}_n$ and $\boldsymbol{\lambda}_n$ are known. Initial guesses for accelerations and Lagrange multipliers are $\ddot{\boldsymbol{x}}_{n+1}^k = \ddot{\boldsymbol{x}}_n$ and $\boldsymbol{\lambda}_{n+1}^k = \boldsymbol{\lambda}_n$, with iteration counter $k = 0$.

2. For the values of $\boldsymbol{\lambda}_{n+1}^k$, we set the sign of corresponding coefficients μ and C_s such that for all contact points. the products $\mu\lambda$ and $C_s\lambda$ are positive. This ensures correct direction for all friction forces irrespective of the sign of individual Lagrangian multipliers.

3. $\dot{\boldsymbol{x}}_{n+1}^k$ and $\boldsymbol{x}_{n+1}^k$ are determined using (3.17) and (3.18). For the positions $\boldsymbol{x}_{n+1}^k$, an iterative Gauss-Newton correction is performed to satisfy the constraint equations $\boldsymbol{C}_k$. While the velocities are corrected in a single step. Both corrections make use of Moore-Penrose pseudo-inverse [16].

4. The quantities quantities are used in (3.19). When $\ddot{\boldsymbol{x}}_{n+1}^k$ and $\boldsymbol{\lambda}_{n+1}^k$ are not exact solutions the right hand side will not be zero. These are the residuals $r_{\ddot{\boldsymbol{x}}}^k$ and $r_{\boldsymbol{\lambda}}^k$. If maximum absolute value of residuals is lower than desired tolerance, $\ddot{\boldsymbol{x}}_{n+1}^k$ and $\boldsymbol{\lambda}_{n+1}^k$ are accepted as the solutions at t_{n+1}. We move to the next time step, starting from step 1. If not, proceed to next step.

5. Correct accelerations and Lagrange multipliers

$$\begin{aligned} \ddot{\boldsymbol{x}}_{n+1}^{k+1} &= \ddot{\boldsymbol{x}}_{n+1}^k + \Delta\ddot{\boldsymbol{x}}^k, \\ \boldsymbol{\lambda}_{n+1}^{k+1} &= \boldsymbol{\lambda}_{n+1}^k + \Delta\boldsymbol{\lambda}^k, \end{aligned}$$

   with the correction values $\Delta\ddot{\boldsymbol{x}}$ and $\Delta\boldsymbol{\lambda}$ given by

$$\begin{pmatrix} \Delta\ddot{\boldsymbol{x}}^k \\ \Delta\boldsymbol{\lambda}^k \end{pmatrix} = -\begin{pmatrix} (1-\alpha_m)\boldsymbol{M}_{n+1} & (1-\alpha_f)(\boldsymbol{C}_{,\boldsymbol{x}}^T - \boldsymbol{W})_{n+1} \\ (1-\alpha_m)(\boldsymbol{C}_{,\boldsymbol{x}})_{n+1} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} r_{\ddot{\boldsymbol{x}}}^k \\ r_{\boldsymbol{\lambda}}^k \end{pmatrix}.$$

6. Increment k and repeat steps 2 to 5 while maximum absolute value of residuals is greater than considered tolerance or until k_{max}.

The parameters of numerical damping γ, β, α_m and α_f are given by Eqn. (17), (20) and (25) in [17]. With these values the Generalised-α method achieves unconditionally stability (in linear regime), second order accuracy and minimal low frequency dissipation. For computing α_m and α_f from (25) of [17], we choose $\rho_\infty = 0$, which means any high frequency response should be eliminated after one time step. We have then set a limit for the number of iterations $k_{max} = 20$ and tolerance for the residuals is 10^{-12} for both accelerations and Lagrange multipliers. Finally, the initial accelerations and multipliers at $t_n = t_0$, $\ddot{\boldsymbol{x}}_0$ and $\boldsymbol{\lambda}_0$ are determined by solving (3.16).

4

Simulation Setup

This chapter goes into detail regarding the platform for the real-time simulation that makes use of the equations of motion and integration algorithm described in the previous chapter. Following which the estimation of the various parameters and validation of the platform is discussed.

4.1. Input Device

The input device that allows one to interact with simulation is an Xbox One game controller. We read the positions of the two joysticks along their respective x axes which are normalised to be between -1 and $+1$. Figure 4.1 below shows the controller along with the considered joysticks. The output of joystick axes are preferred to those of buttons as the output of the latter is binary, either a 0 or 1. The x position of joystick 1 is considered for M_α that represents shoulder and knee steering force, while that of joystick 2 is considered for head steering moment M_{HS}.

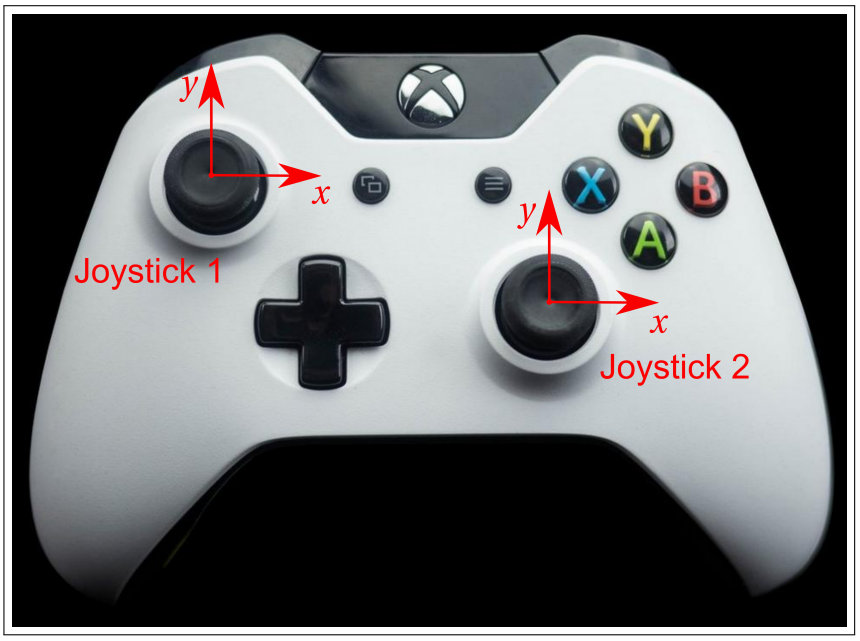


Figure 4.1: The game controller used as steering input device. The figure shows the two joysticks along with considered axes. x axis of joystick 1 acts as input for shoulder and knee steering through M_α while that of joystick 2 acts as input for head steering through M_{HS}.

4.2. Visualization and Output Device

The code for the numerical integration and most of the track reconstruction was written in MATLAB as it offers advantages for vectorized operations that are performed. Unfortunately, MATLAB is not well suited for real-time graphics rendering. This task is handled by Unity game engine. A TCP/IP network connection is established between MATLAB and Unity with Unity as the server and MATLAB as the client. After each time step of the numerical integration, the position and orientation of athlete head location is sent from MATLAB to Unity. Athlete head orientation is that of frame $\mathcal{B}$ w.r.t $\mathcal{N}$. From Sec. 3.1, the position of athlete head is estimated as

$$\boldsymbol{p}_A = \boldsymbol{p}_S + {}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}} \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix}. \tag{4.1}$$

This position and orientation is used to control that of a virtual camera in an environment that contains the track surface. The position and orientation of the camera is updated at a rate of 30 Hz. For the simulated runs performed, the output device is a stationary screen as shown in Fig. 4.2 below.

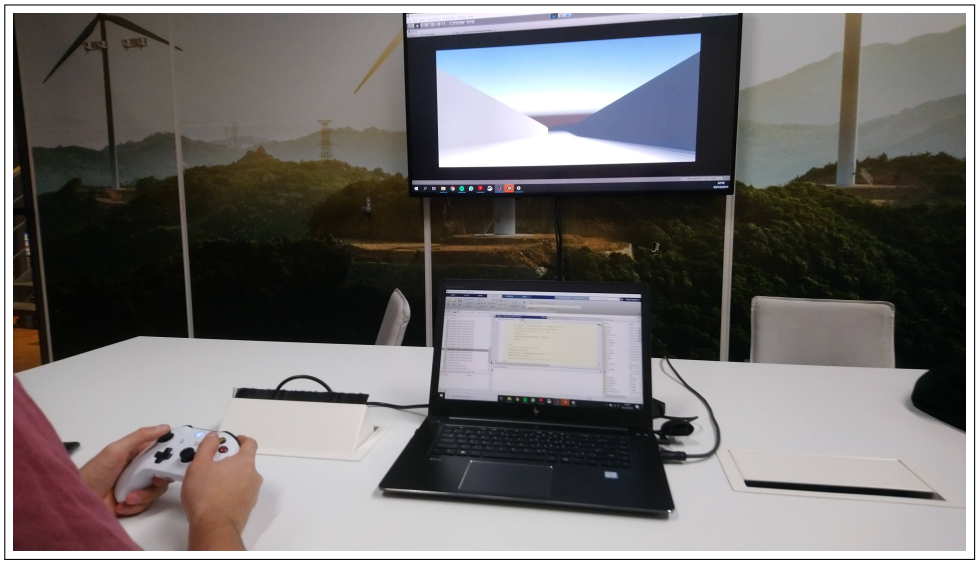


Figure 4.2: The simulation setup. On the left, a user provides steering input using a controller. The computation is performed in MATLAB on the laptop in the centre. The position and orientation on track is displayed in Unity on the screen in the background.

4.3. Parameter Estimation

The equations of motion described in Chapter 3, contain a large number of parameters whose values must be appropriately estimated for a successful simulation. The reasoning behind this will be explained here, the parameters will appear in the same order as their appearance in Chapter 3.

Sled Model Parameters

The mass of the sled m, is assumed to be 115 kg, which is the maximum permissible limit for combined mass of athlete and sled for participation in IBSF competitions [18]. Most athletes perform runs close to this maximum limit by adding additional weight on to the sled. The dimensions of the sled is assumed to be that of just the athlete. The athlete chosen for the simulation is Akwasi Frimpong from Ghana, having a height h of 1.73 m, this was obtained from the International Bobsleigh and Skeleton Federation website[1]. The remaining two dimensions w and d were approximated to be 0.45 m and 0.23 m respectively [19].

The distance between two contact points on the same runner was initially chosen to be 0.035 m based on the results of the model formulated in [20]. Thus, $a = 0.0175$ m. The distances b and c are chosen based on the restrictions set by IBSF rules for Skeleton [18]. The distance between two runners along the lateral Y' axis

[1] `https://www.ibsf.org/en/athletes/athlete/263856/Frimpong` Retrieved 01/11/2019

of the body fixed frame is restricted to be between 0.340 m and 0.380 m. For our model, we take the mean of these two distances resulting in $b = 0.180$ m. The frame on the sled upon which the athlete rests is between 0.08 to 0.12 m above the track surface. As the body is the combined athlete and sled body, we take $c = 0.125$ m. The moment of inertia associated with generalized coordinate α is initially assumed to be zero.

Coefficients of Friction

The coefficient of friction between runners and ice surface is itself dependent on a number of parameters. Here will consider only the differences that arise due to the changes in contact geometry between the front half and rear half of a Skeleton runner. The geometry front half of the runners is identical to that of a Bobsleigh runner [20]. This coefficient is chosen to be $\mu_{front} = 4 \times 10^{-3}$. The value was used by Rempfler and Glocker for their Bobsleigh simulator [6] and is considered to be on the lower side of previously reported values for Bobsleigh runners.

The geometry of the rear half of the runner differs significantly, more closely resembling that of a speed skating blade. There has been one previous attempt to quantify it and the maximum reported value is 5.7×10^{-3} [20]. This is at the higher end of the values reported by de Koning et al. [21] for speed skate blades. To begin with, we take $\mu_{rear} = 6 \times 10^{-3}$.

Cornering Stiffness

The magnitude of a steering force given by Eqn. (3.6) is,

$$f_L = C_s \delta |\lambda|, \text{ or}$$
$$\frac{f_L}{|\lambda|} = C_s \delta.$$

The steering force f_L can not be larger than the contact normal force $|\lambda|$, therefore

$$C_s \delta \leq 1, \text{ or}$$
$$C_s \leq 1/\delta$$

The maximum value of the steering angle δ is estimated to be 0.0873 rad. (5°). The value is half of the maximum allowable steering angle for a particular Bobsleigh construction [6]. Given that the runners of a Skeleton sled cannot be rotated like those of Bobsleigh, it is not expected that the an angle greater than this will be encountered during a descent. Thus,

$$C_s \leq 11.5$$

Aerodynamic Drag Parameters

This drag force is dependent on three parameters. First, the coefficient of drag C_D, we take this to be 0.38, which was used by Roche et al. in their model for Skeleton [5]. Next, the frontal area A_F is assumed to be equal to the rectangular area formed by the dimensions w and d of the athlete. Finally the density of air ρ is 1.275 kg/m^3.

Integration Time Step

The time step for the simulator was chosen based on the average iteration times for 2 seconds of simulated descents on a flat and straight segment. This mean iteration time was usually close to 0.5×10^{-3} s and hence we use a time step, $\Delta t = 0.01$ s.

4.3.1. Initial Conditions

Athletes begin steering once they jump on to the sled close to the 35 m mark. We take this as the starting position, with a speed of 40 km/hr. The position is corrected using a Gauss-Newton algorithm and subsequently the velocity is corrected to ensure that they satisfy the constraints $\boldsymbol{C}_k$. The initial accelerations $\ddot{\boldsymbol{x}}_0$ are then determined by solving Eqn. (3.16).

4.3.2. Trial Simulations

With the parameters and initial conditions above, attempts were made to replicate the path taken as seen in the accompanying video[2]. First observations were that lateral movement of the sled was larger than expected while the amount of rotation about Z' axis was lesser than observed in the video. The coefficient for the rear contact points was tuned to by a factor of 2 to be $\mu_{rear} = 12 \times 10^{-3}$. While the cornering stiffness parameter was reduced by a factor of four. Thus $C_s = 2.5$.

The second preliminary observation was oscillations in the values of Lagrangian multipliers as the sled traverses a curved segment. To mitigate this, first we tuned the moments of inertia ${}^{\mathcal{B}}I_{yy}$ and I_α to be

$$
\begin{aligned}
{}^{\mathcal{B}}I_{yy} &= 0.675 \times \frac{1}{12} m(h^2 + d^2), \\
I_\alpha &= 0.325 \times \frac{1}{12} m(h^2 + d^2).
\end{aligned}
$$

Second, the distance between contact points on the same runner was tuned to $a = 0.175$ m. The differences are shown in Fig. 4.3 and 4.4.

Scaling M_α & M_{HS} Steering Input

The position of the joystick is normalised to be between −1 and +1 and has to be scaled appropriately. To decide the maximum value of M_α for joystick 1, we observed the effect that a linear increase of its value from 0 to 200 has on the yaw acceleration during traversal of a straight flat section of track. This is shown in Fig. 4.5. We saw that M_α has a maximum effective value beyond which further increases do not increase the yaw acceleration. This happens when the magnitude of the two Lagrange multipliers that are reduced as a consequence of M_α (see Fig. 3.4) reach zero. Further increase in M_α, leads to a change in sign and increase in magnitude. This means there is no further increase in difference of magnitudes between opposite side multipliers (Left Front vs. Right Front & Left Rear vs. Right Rear). We actually see small decreases in the difference between the rear contact points as M_α increases further. We attempt to illustrate this in Fig. 4.6. This occurred close to M_α increasing past 100 and we take this as the limit. For M_{HS} we took an arbitrary maximum magnitude of 10.

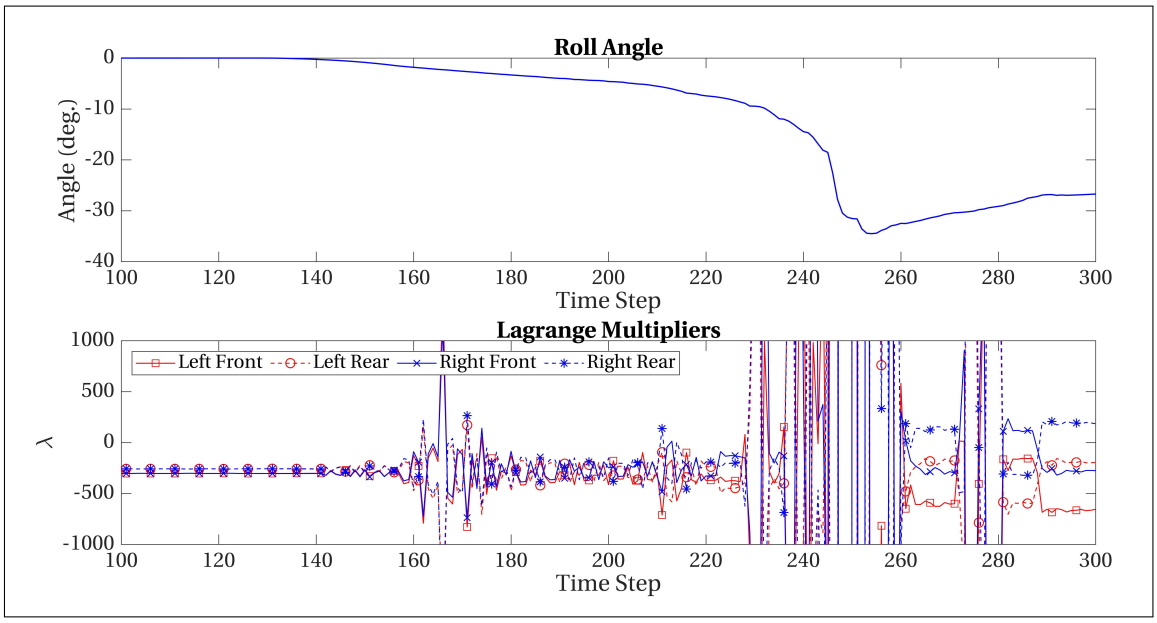


Figure 4.3: The above plots show the oscillations of Lagrangian multipliers as the body rotates during traversal of curve, with zero steering action and parameters $a = 0.0175$, ${}^{\mathcal{B}}I_{yy} = \frac{1}{12} m(h^2 + d^2)$, $I_\alpha = 0$.

[2] `https://www.youtube.com/watch?v=bgPEQ4rWLNI` Retrieved 01/11/2019

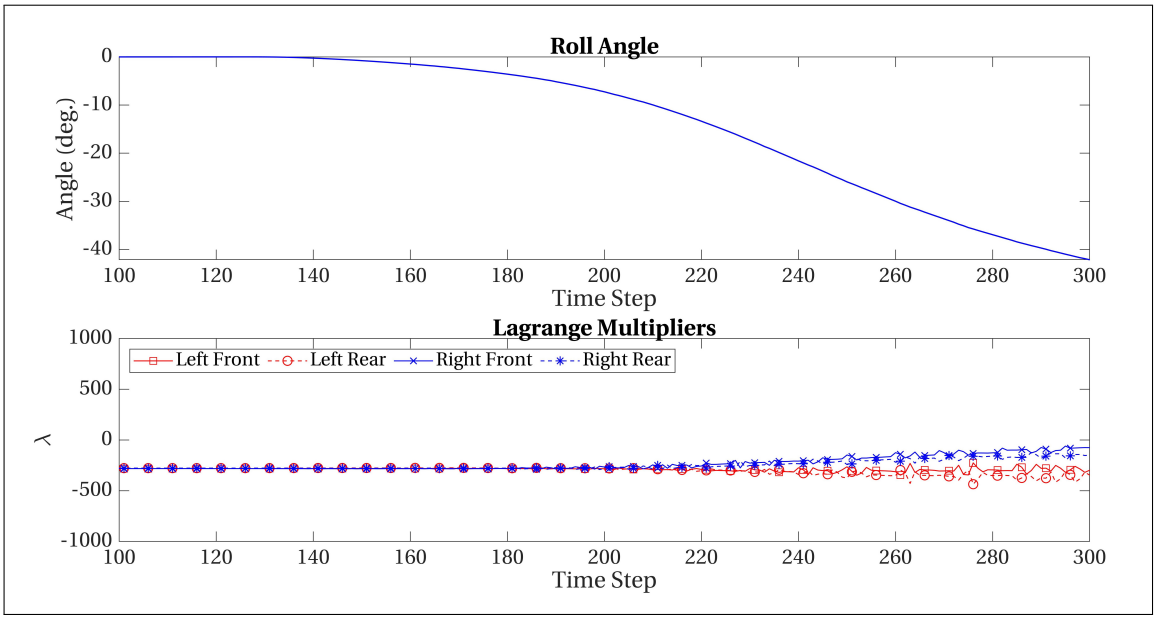


Figure 4.4: The above plots show the Lagrangian multipliers as the body rotates during traversal of curve 1, with zero steering action. The effect of changing the parameters to a, ${}^{\mathcal{B}}I_{yy}$, I_{α} can be clearly seen when compared to Figure 4.3

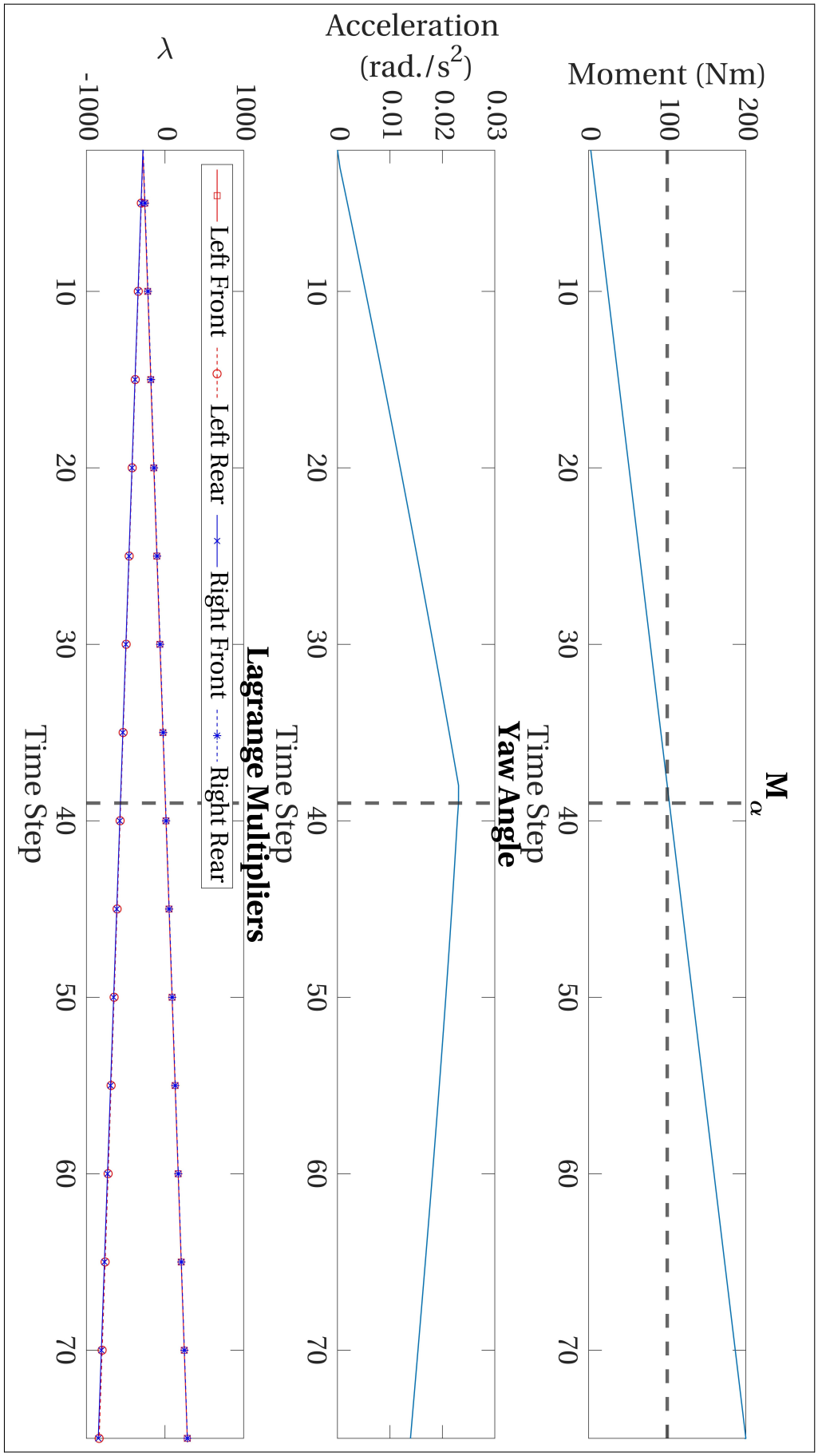


Figure 4.5: The effect of M_α on yaw acceleration and Lagrange multipliers. The vertical dashed lines are at time step 39, beyond which further increases in M_α are no longer effective. The horizontal dashed line in the first plot at $M_\alpha = 100$, is the limit for controller input we have chosen based on the above observation.

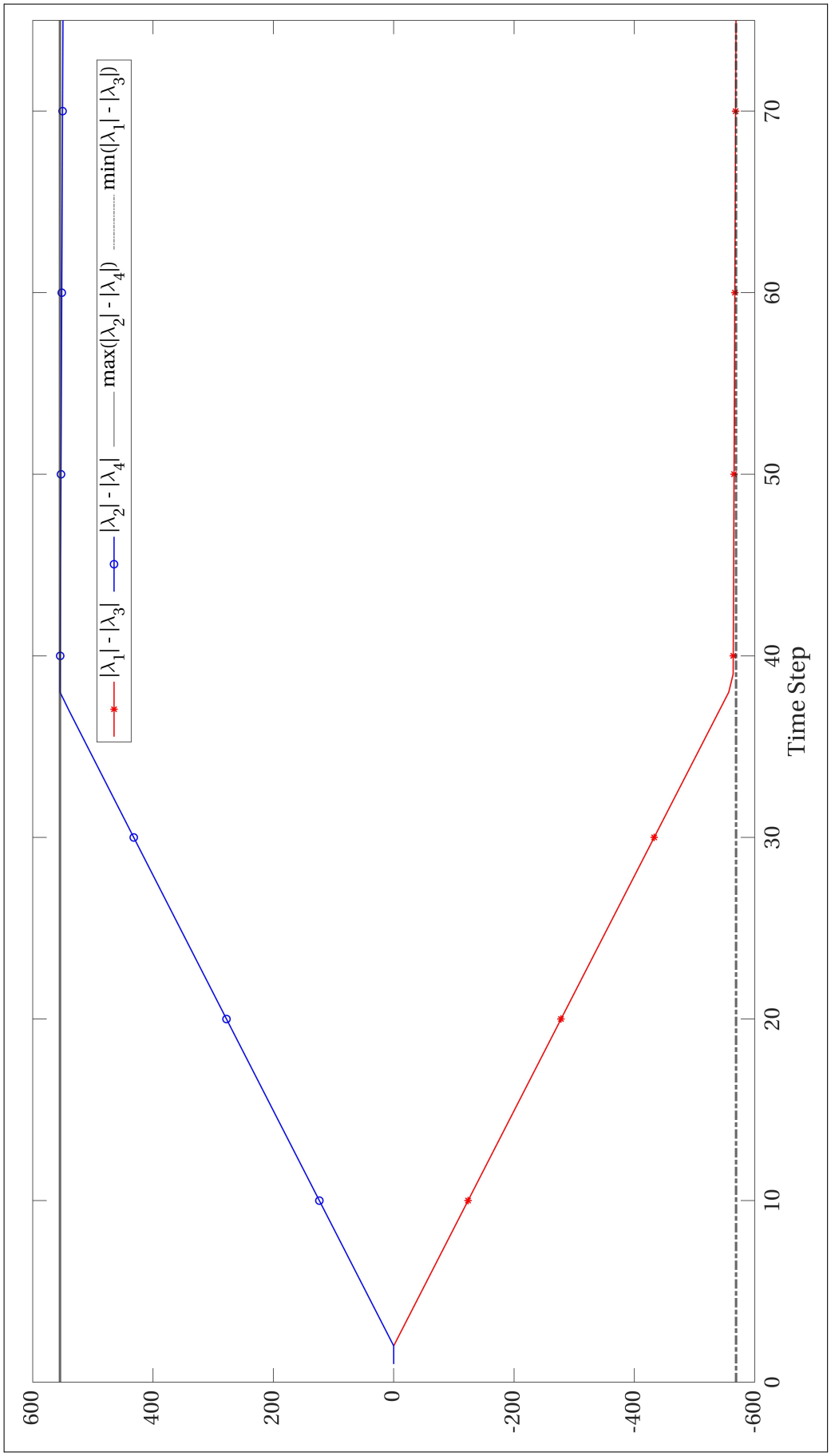


Figure 4.6: The changes in differences between opposite sided Lagrangian Multipliers as M_α increases from 0 to 200. The magnitude of difference between λ_1 (Left Front) and λ_3 (Right Front) continues to increases ever so slightly. While, the magnitude of difference between λ_2 (Left Rear) and λ_4 (Right Rear) decreases slightly.

5

Results and Discussions

In this chapter we examine the results of simulated runs. We initially look at paths traversed and computation times before making more detailed observations. Finally, we attempt to validate the simulator using time elapsed for traversal of certain segment of the track and the sum of Lagrangian multipliers.

5.1. Results

First attempts were made at real-time, the furthest distance travelled was past the exit of curve 2. We then proceeded to run the simulator at a slower speed, 4 times slower to be exact. At this play speed we were able to reach the end of curve 4 consistently. We present the results of five runs that were longest in terms of duration. The paths traversed can be seen in Fig. 5.1. In Table 5.1 we show the total duration and mean of computation times for each time step of these runs.

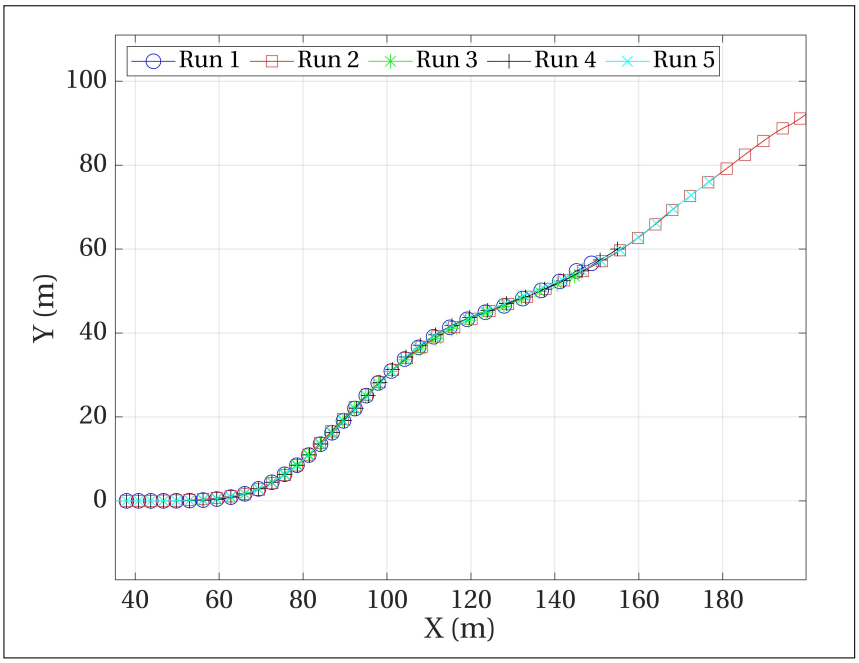


Figure 5.1: Paths traversed in XY plane during simulated runs at real-time speed.

Table 5.1: Total duration of and mean computation time per time step for real-time simulation runs

Run #	1	2	3	4	5
Total Duration (s)	8.59	11.36	8.37	8.82	10.14
Mean Computation Time (s)	0.0060	0.0064	0.0064	0.0063	0.0068

We carried out 15 runs at a slower speed. Here again we restrict our results to the five runs that last the longest in terms of duration. The path traversed is illustrated in Figures 5.2 and 5.3. While computation times are presented in Table 5.2. The simulation speed is reduced using 'pause' command in MATLAB at the end of the computation of a time step and hence we can still determine the computation time for each time step.

Table 5.2: Total duration of and mean computation time per time step for slower simulation runs

Run #	1	2	3	4	5
Total Duration (s)	15.99	16.39	17.05	17.82	18.25
Mean Computation Time (s)	0.0063	0.0061	0.0065	0.0062	0.0059

Next, we plot the (absolute of the) sum of magnitudes of contact Lagrangian multipliers for these 5 simulations in Figure 5.4. The rolling action during traversal of curve 4 can be seen in the reduction of these sums towards the end. In both Runs 4 and 5, the user was able to reach the exit of curve 5 before losing control and subsequent stoppage of the simulation. We choose Run 4 for further investigation as both traversed the same distance as seen in Fig. 5.2 but 4 did it quicker than run 6.

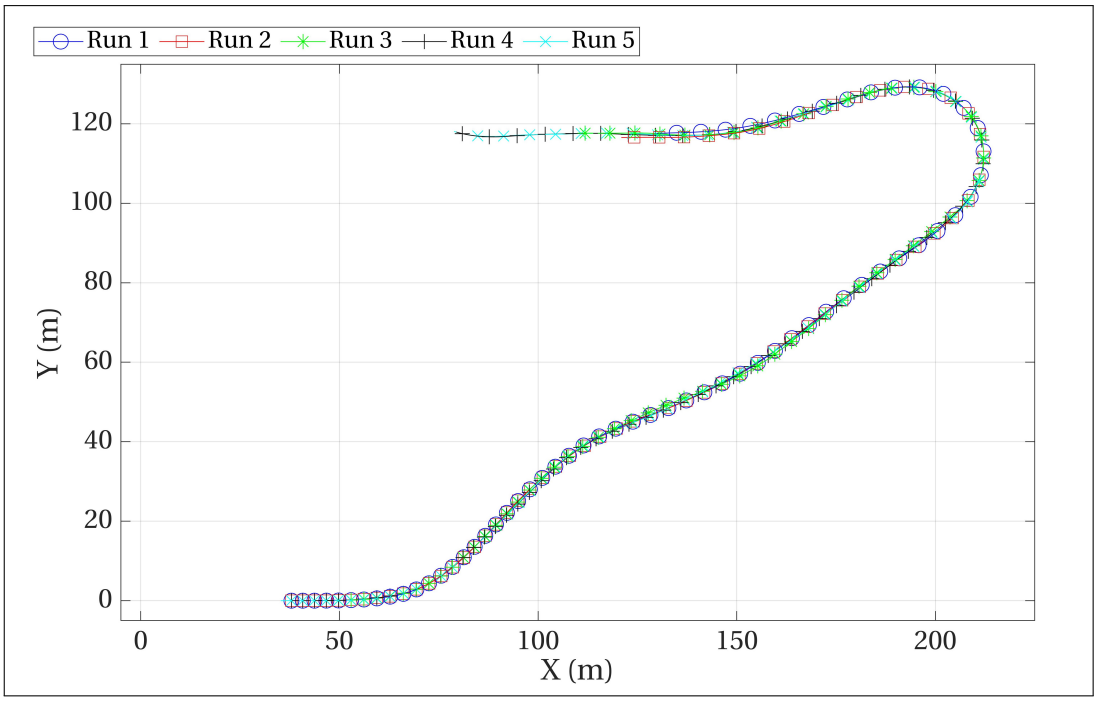


Figure 5.2: Path traversed in slower simulation runs in XY plane.

For Run 4 we present more data, starting with the steering input given by the user in Fig. 5.5. Followed with linear velocity and accleration in Fig. 5.6 and 5.7. We then proceed to show the orientation of sled frame $\mathcal{B}$ w.r.t. Inertial frame $\mathcal{N}$ in terms of Euler angles using a Yaw-Pitch-Roll convention along with velocities and accelerations of these angles. This is shown in Fig. 5.8. Finally we show the change in energy per time step in Fig. 5.9 and the angle δ for all four contact points in Fig. 5.10.

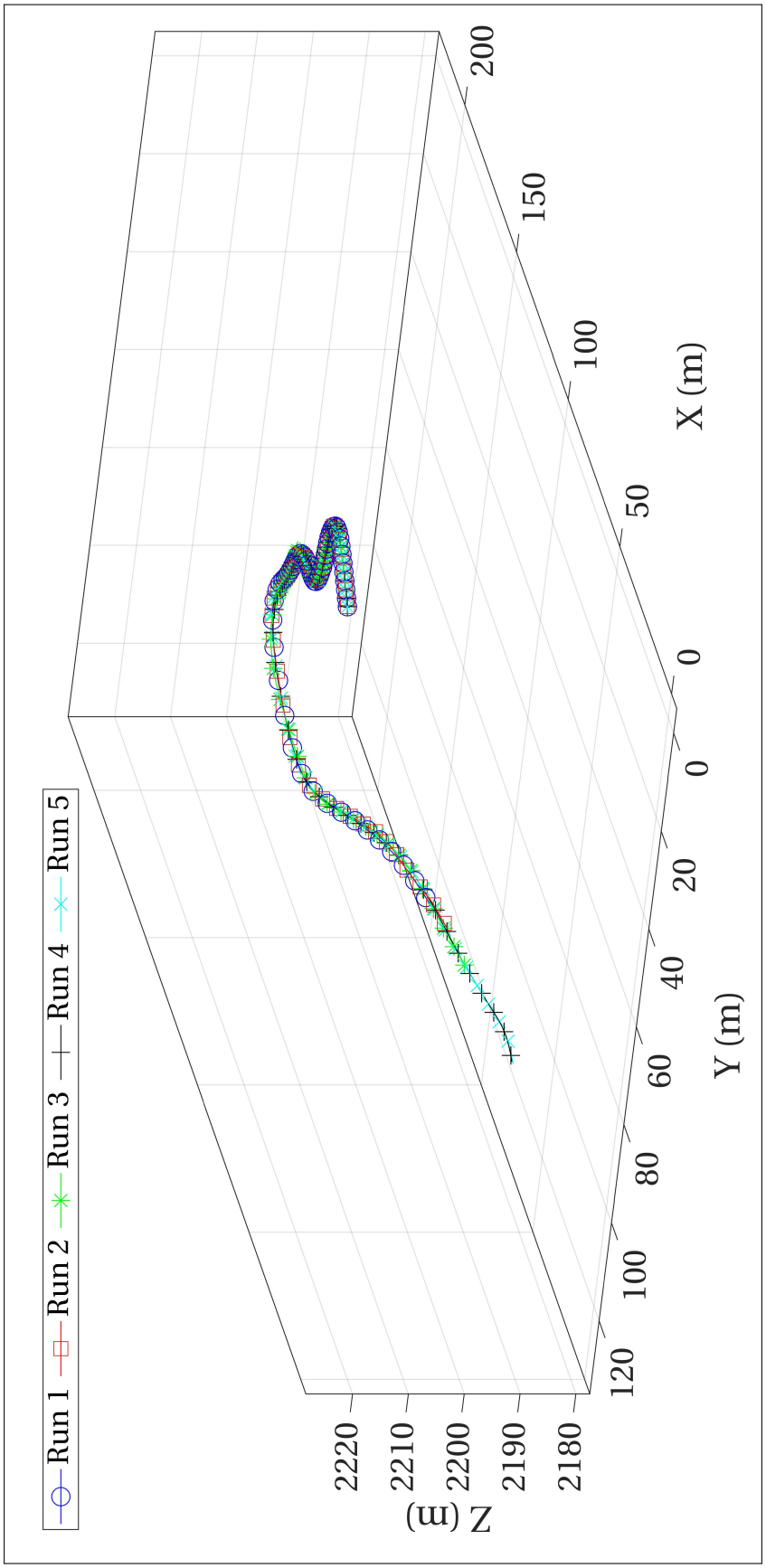


Figure 5.3: Path traversed in slower simulation runs in 3D space.

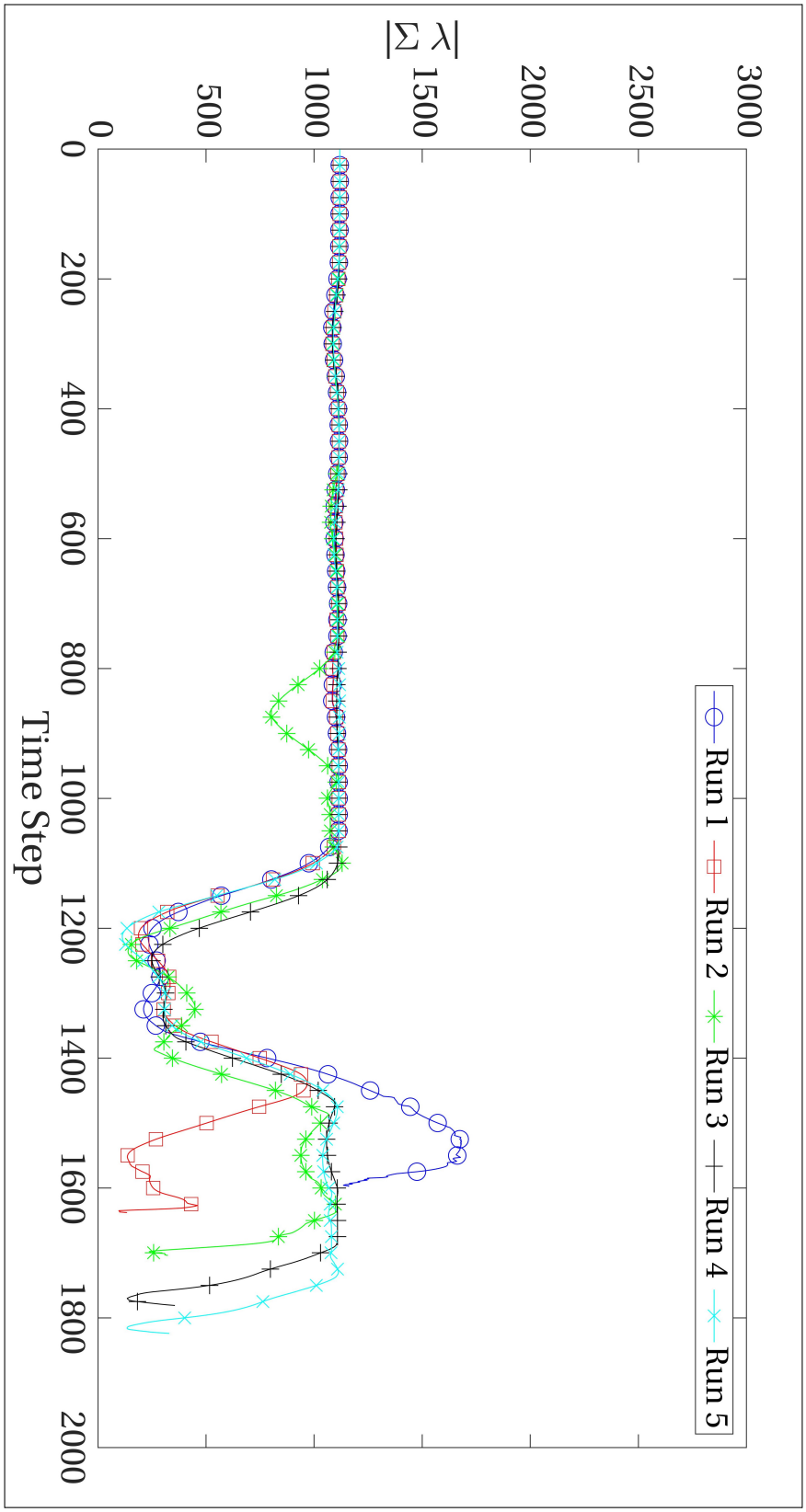


Figure 5.4: Absolute value of sum of contact Lagrangian multipliers for simulations run at lower speed.

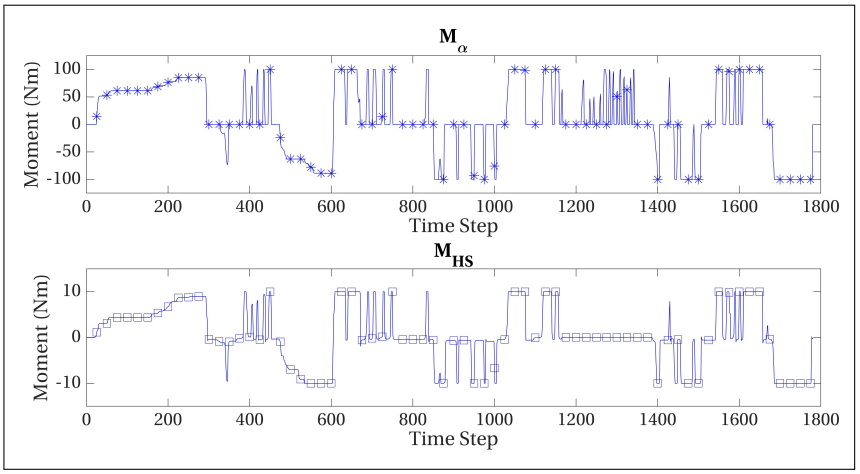


Figure 5.5: Steering Input given by user for Run 4

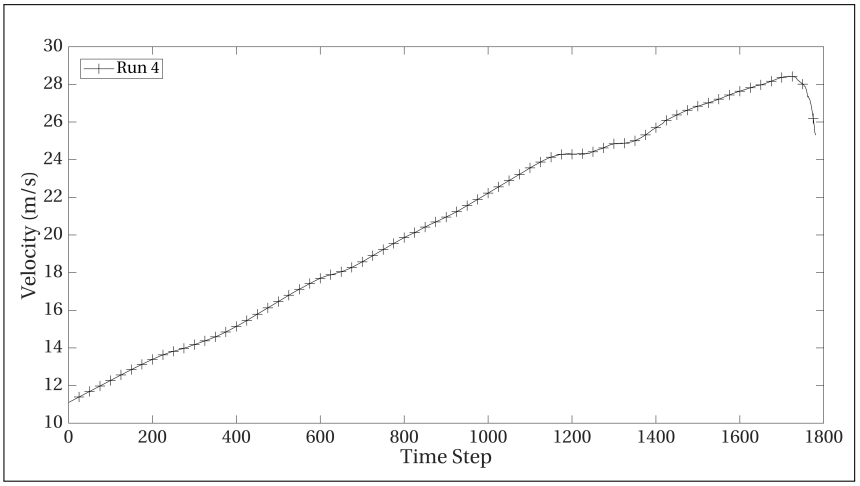


Figure 5.6: Linear Velocity of centre of mass *S* for Run 4 of slower descents.

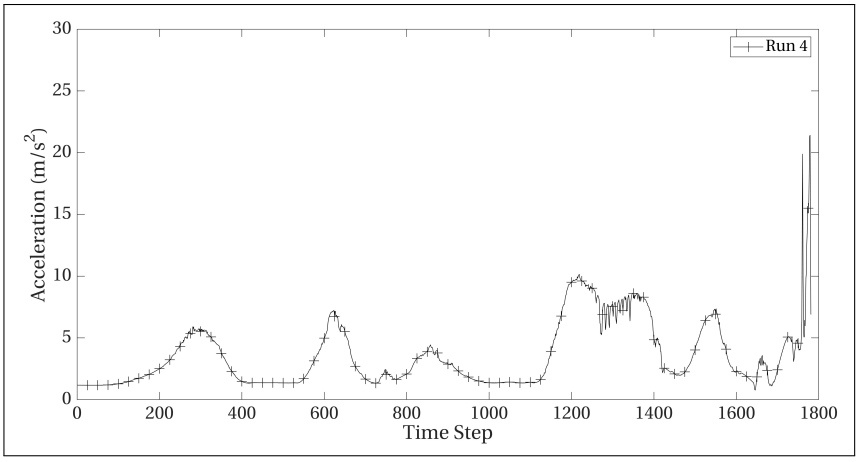


Figure 5.7: Linear Acceleration of centre of mass *S* for Run 4 of slower descents. Unusually high acclerations are seen close to stoppage of the simulation.

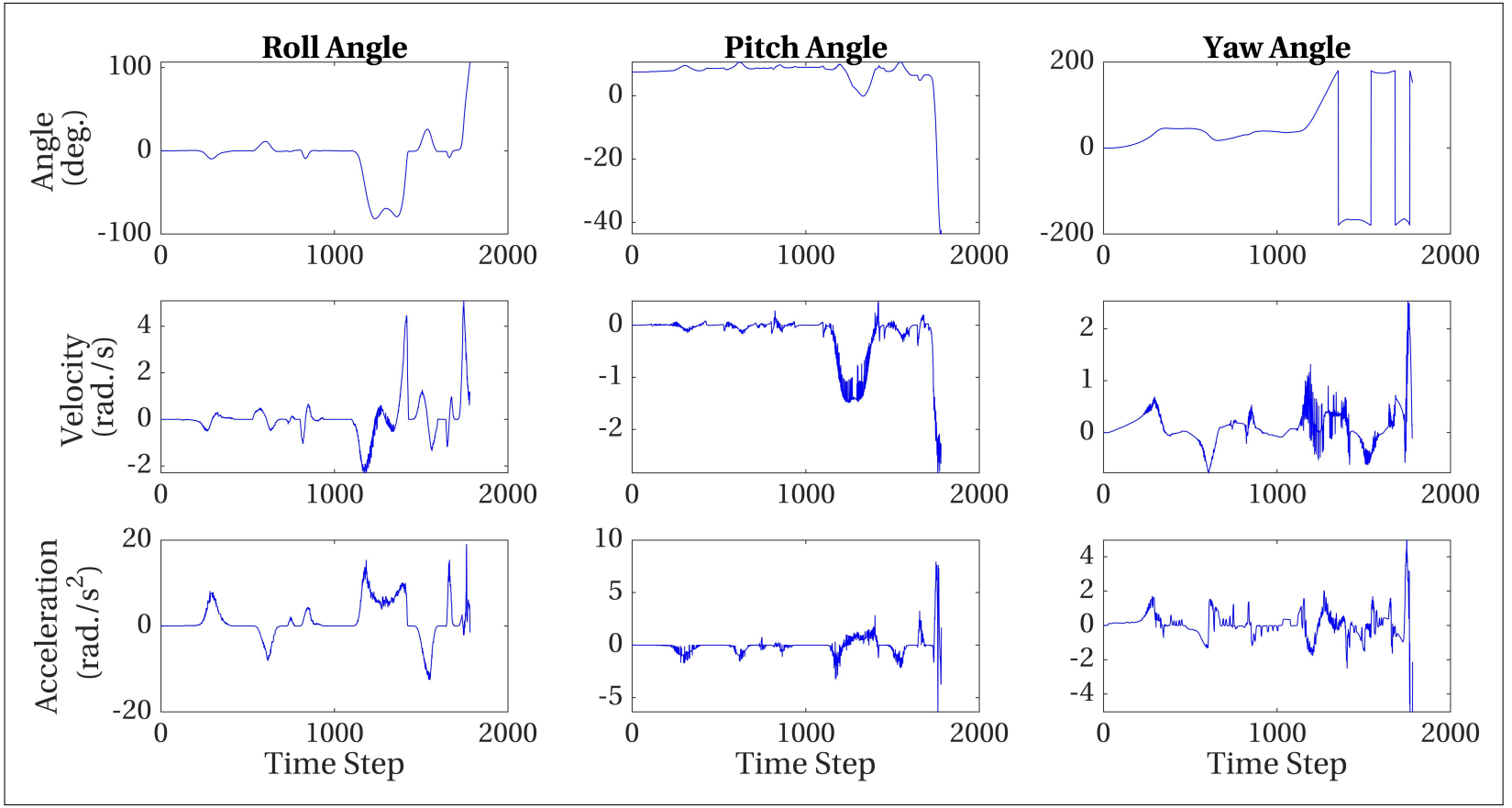


Figure 5.8: Angular quantities for Run 4 of slower speed simulations. Like in the case of linear accelerations in Fig. 5.7 high accelerations of angular quantities can be seen close to stoppage of the simulation.

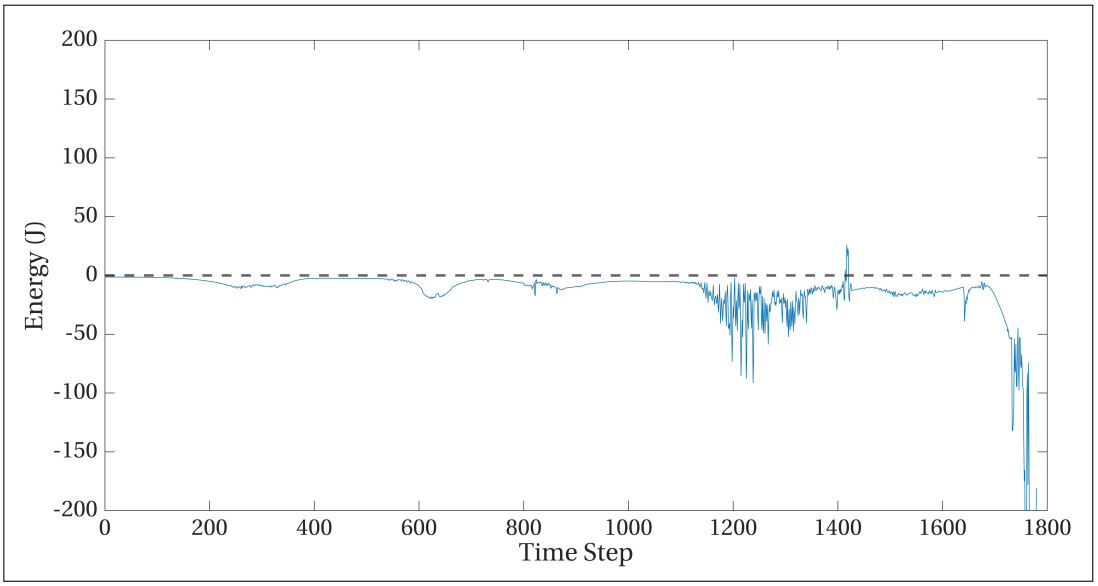


Figure 5.9: Change in Energy during Run 4. Excluding a peak close to the 1400th time step, energy is dissipated at every time step.

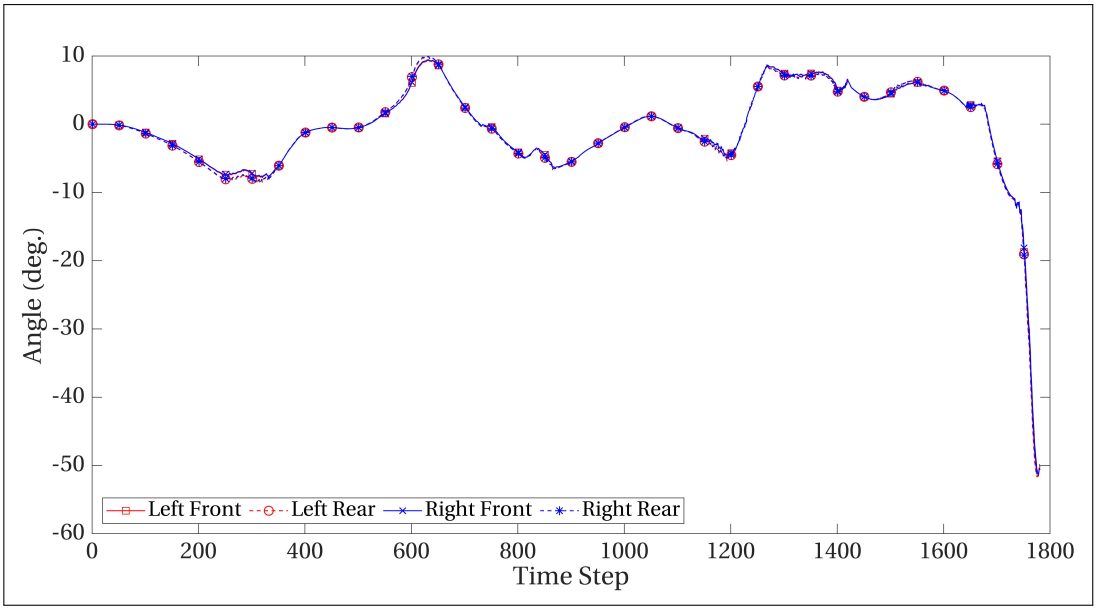


Figure 5.10: Steer angles of all four contact points during Run 4. For most of the simulation, the angle stays within reasonable bounds. Unexpected high values are seen as the simulation begins to fail.

5.1.1. Validation

Computation Times

In Figure 5.11, we highlight time steps for which the computation time was longer than the time step for the simulations run in real-time. Our initial choice for time step, $\Delta t = 0.01$ s based on computation time on a straight flat section seems to hold even for longer distances travelled and traversal of curves. Out of little more than 4700 steps, 136 took longer than the integration time step. Most of these time steps are clustered towards the end of a run just before the simulator fails. The stoppage is seen with the increase in number of iterations needed to solve the equations of motion (3.19) as seen in Fig. 5.12. The first few time steps also seem to take longer in every run even though only 3 or 4 iterations are needed. The exact cause is unknown.

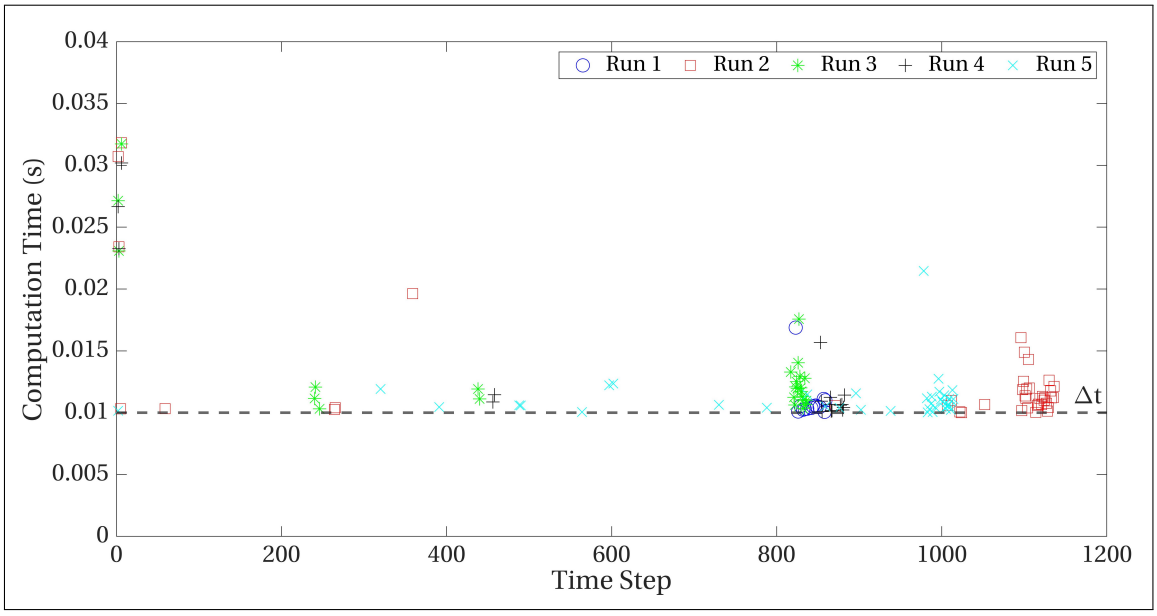


Figure 5.11: The plot shows computation times that were longer than the integration time step Δt for real-time simulations. The longer steps are clustered just before stoppage of the simulation for that run.

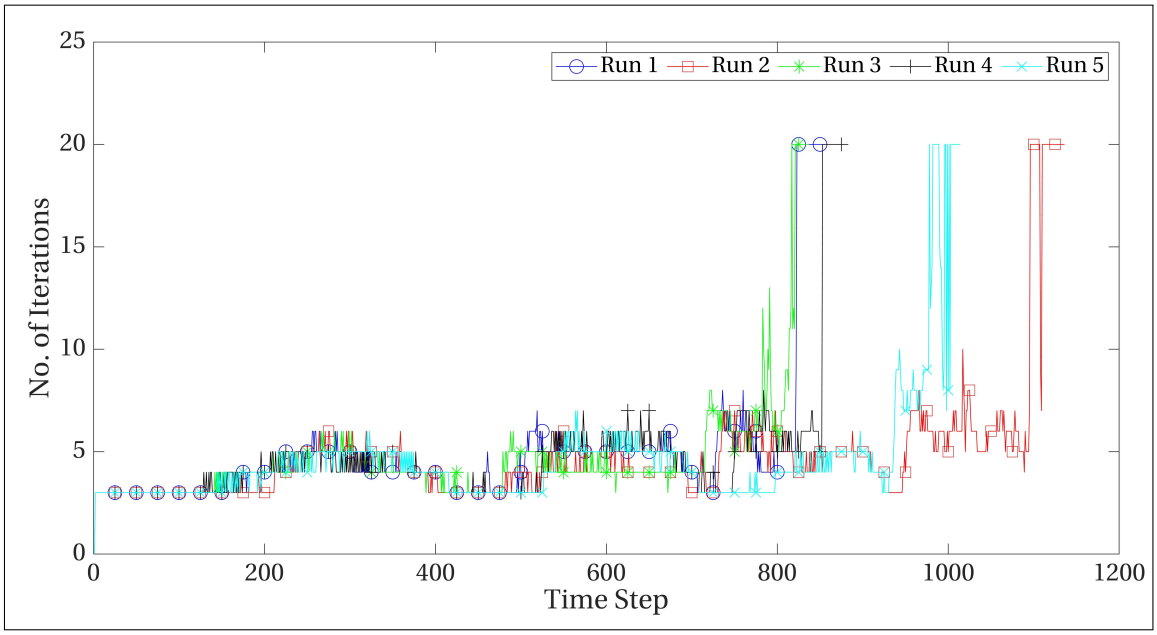


Figure 5.12: The number of iterations taken to solve (3.19) for real-time simulations. which increases to the maximum 20 as the simulation begins to fail.

For the slower simulations, we show the same computation time data for Run 4 in Fig. 5.13 & 5.14. We see similar trends to the real-time simulations.

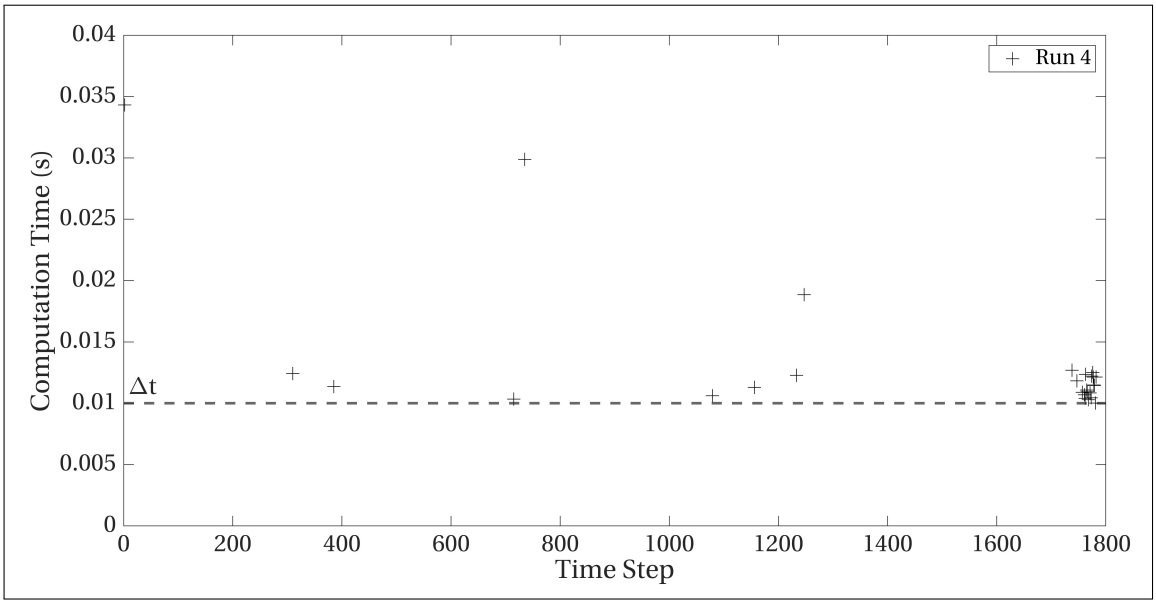


Figure 5.13: The plot shows computation times that were longer than the integration time step Δt. The longer steps are clustered just before stoppage of the simulation for that run. This considers Run 4 of the slower speed simulations.

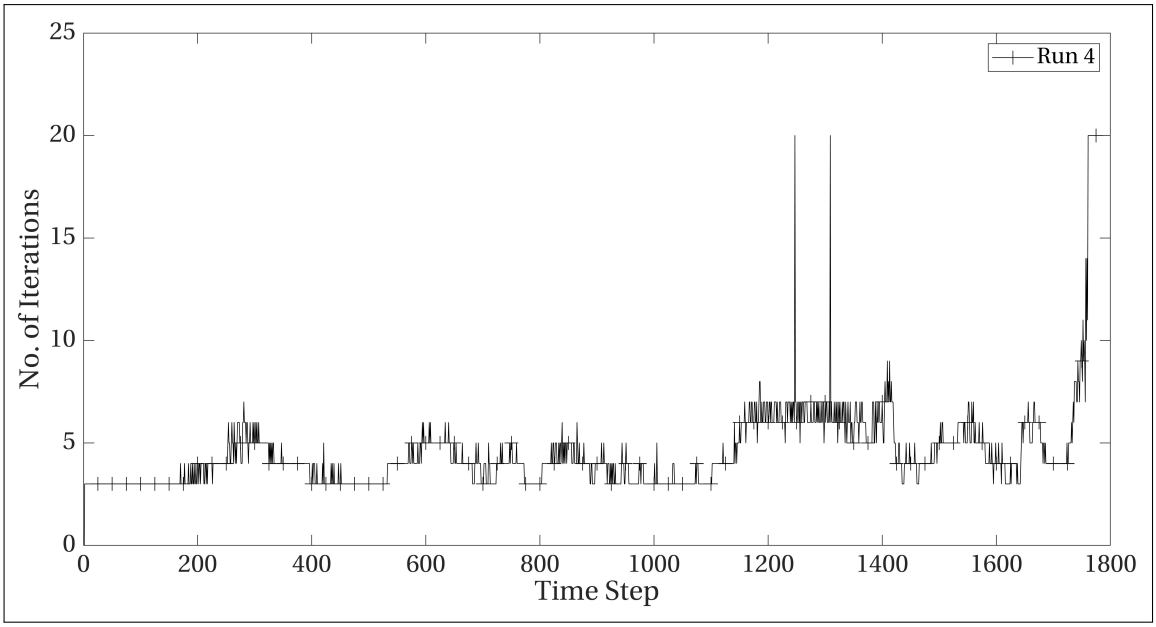


Figure 5.14: The number of iterations taken to solve (3.19) for Run 4 of slower speed simulations. which increases to the maximum 20 as the simulation begins to fail.

Elapsed Time

Next, we compare time elapsed during traversal of a segment of the track in simulated runs to those of actual races. At the Park City Utah track, the first split time is recorded at the entry to curve 1 and the second time is recorded at the entry to curve 6. Since we have reliable data only up to curve 4. We consider the time taken from split 1 to entry of curve 4. We take the mean times of the first three competitors from footage available for Heat 1 of Men's Skeleton event at the IBSF World Cup during the 2017/18 season [1]. To extract this time from the simulated runs we determine the number of time steps taken from entry of curve 1 to entry of curve 4. Based on the divsion of segments and interpolation between cross-sections in Sec. 2.1.3, entry to curve 1

[1] `https://www.youtube.com/watch?v=hMz8q5syGrE&t=378s`

is approximated to when the centre of mass crosses the set of triangles formed by the 500^{th} and 501^{st} cross-sections, while entry to curve 4 is when it crosses those formed by 2800^{th} and 2801^{st} cross-sections. For this we consider the simulated runs that were run at slower speed. These times along with speed at the entry to curve 1 are shown in Table 5.3.

Table 5.3: Comparison of Time taken to travel from entry to curve 1 to entry to curve 4 and speeds at entry to curve 1 between simulated descents and actual descents.

	Mean Time - Curve 1 to 4 (s)	Mean Speed - Entry to Curve 1 in (m/s)
Competition descents	9.127	12.611
Simulated runs	10.534	12.470
Percentage Difference	15.41%	1.12%

Sum of Lagrange Multipliers

The sum of magnitude of Lagrangian multipliers provides a second means to validate the model. We compare this sum to the weight of the sled body acting normal to the track surface for Run 4 in Fig. 5.15. The weight of the sled normal w_S to surface is computed using

$$w_S = m|g|\cos\phi. \tag{5.1}$$

where, m is the mass of combined athlete-sled body, g is the acceleration due to gravity and ϕ is the roll angle.

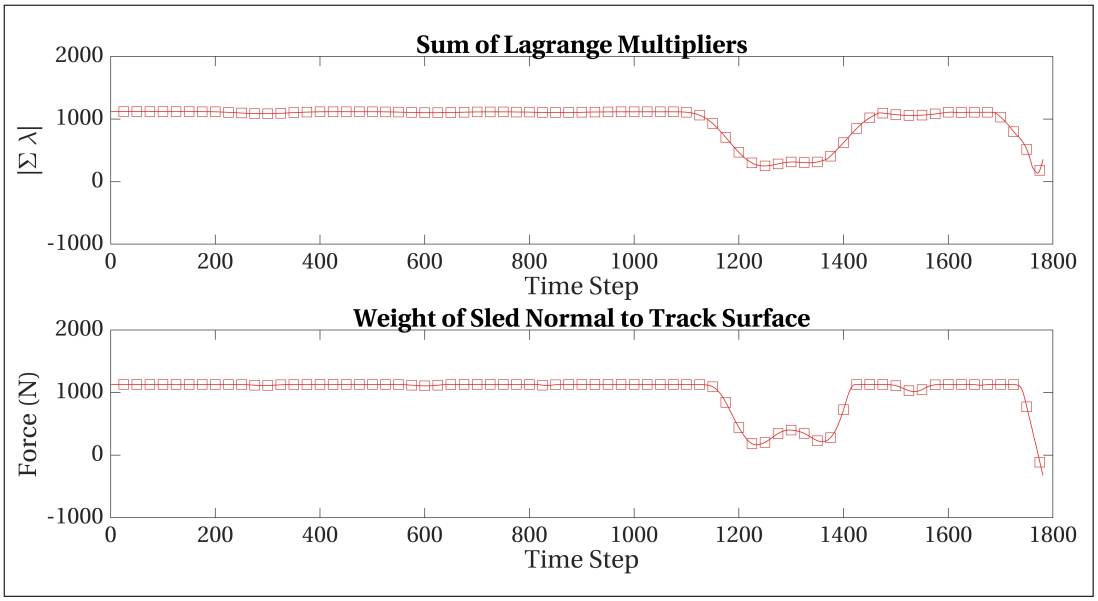


Figure 5.15: (Absolute of) Sum of Lagrange multipliers and weight of sled normal to track surface for Run 4.

5.2. Discussions

5.2.1. Validation

Computation Times

For most of the simulated time steps, the algorithm (Section 3.5.1) proves to be fast enough. For the runs simulated at real-time speeds, out of around 4728 time steps, around 136 took longer than the integration time step Δt chosen. This is around just 2.87 %. We look closer at the computation times for Run 4 of the slower simulations. In total 27 steps took longer to be computed than chosen Δt. Out of these 18 are clustered towards the end of the simulation. If we were to neglect them, just 9 of the 1700-odd time steps took longer, which is just 0.016 %.

We see a minor correlation in the number of iterations needed to solve the equations (3.16) and traversal of curved segments. To illustrate this we plot the number of iterations against the roll angle in Figure 5.16.

The roll angle serves as an approximation for radius of curvature as it tends to increase as radius of curvature of segment reduces. On first glance it may appear that high angles cause problems in solving the equation of motion but we look back to fig. 5.14 we see that most the 20-iteration time steps occur towards the end just before stoppage of the simulation. We believe it is a good sign that the number of iterations does not increase drastically with the roll angle. For flat sections it appears that the algorithm requires between 3 to 4 iterations.

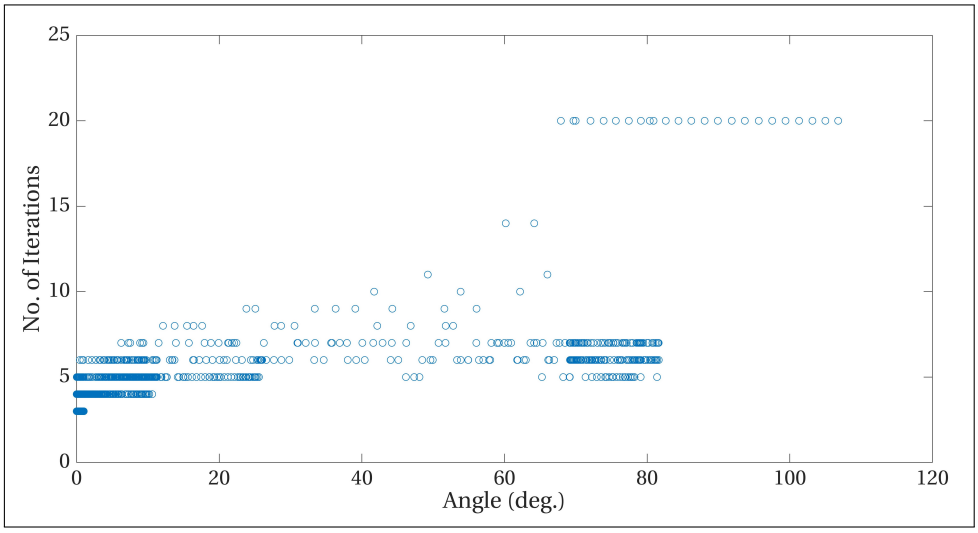


Figure 5.16: Number of iterations as a function of roll angle for Run 4

Elapsed Time
In Table 5.3 see that the simulated runs take around 1.5 seconds longer. There are multiple reasons for the mismatch. First, are errors in the approximation of time elapsed for simulated runs and time at entry to curve 4 in the actual descents. Second we don't believe the path taken during the simulated runs are optimal, there are impacts with side wall in most runs. Finally, either the damping introduced through the time stepping scheme (Section 3.5.1) or the increase in rear contact point friction coefficients (from 0.006 to 0.012) could be too high. The difference of 15% is quite high for a sport where the winning margins are fractions of a second. Even so, whether this slower descent time actually makes the simulator less effective as training tool is a question that needs to be answered. Not today, but when complete descents are possible.

Sum of Lagrange Multipliers
In Figure 5.15, we see that the sum of Lagrange Multiplier values closely follows that of the weight of sled normal to track surface. The two troughs as the sled traverses curve 4 are a result of the rise and fall of the sled, this is a well known phenomenon in the sport [24]. The sled usually will not travel a curve in such a way that there is zero across track movement due to the changes in components of both inertial (centrifugal or centripetal) forces and weight of sled that are normal to track surface.

On the whole we consider the validation a partial success. The algorithm is fast enough, the contact Lagrangian multipliers behave as expected, but the elapsed time compared poorly to real world ones. The ideal scenario is one in which we are able to complete a descent of the entire track and the descent time matching that of real world descents.

5.2.2. Additional Observations

We would like to discuss a few more observations from Run 4. First, the choice of using joysticks over buttons seems inconsequential as the user tends to use them at their respective maximum values as seen in Figure 5.5. This could mean that the effect produced is not enough to steer the sled down the optimal trajectory. Next, there is a drastic increase in loss of energy and extremely high (linear and rotational) accelerations are seen prior to stoppage of the simulation as seen in Fig. 5.7, 5.8 and 5.9, which needs further investigation.

Third, we turn your attention to Fig. 5.10, that shows the steering angle δ for the four contact points dur-

ing Run 4. We see that all four angles have the same sign and nearly the same magnitude for the duration of the simulation. This means that the contribution to $\boldsymbol{v}_{lat}$ (Fig. 3.3) from the linear velocity of S $\boldsymbol{v}_s$, (term 1 of R.H.S. of Eqn. (3.4)) is usually much higher than that of velocity given by ${}^{\mathcal{N}}\boldsymbol{R}_{\mathcal{B}}({}^{\mathcal{B}}\boldsymbol{\omega} \times {}^{\mathcal{B}}\boldsymbol{p}_k)$ (term 2 of R.H.S. of Eqn. (3.4)).

Now we turn our focus towards the experience of using the simulator. The users found it very hard to control the sled in real-time. An obvious reason is that they are not accustomed to the speeds involved with the sport. Slowing down the playback speed of the simulation did help but still a complete descent of the track was not possible. It would be a good idea to have a Skeleton athlete try the simulator, they might be able to provide better feedback based on their experience of sliding down tracks and would naturally be better judges of the simulation.

5.2.3. Track Model

A video of a descent following a pre-determined path down the generated surface was shown to a skeleton athlete, Akwasi Frimpong who regularly trains at the Park City track and positive feedback was received regarding likeness to the actual track. But due to the nature of the source information, mainly the use of video footage to approximate the transverse cross-sections, the surface generated is an approximation of the track surface. First, using 2D images to approximate a line in 3D space leads to errors. Second, the decision to assemble the cross-sections in curved segments equally distant from each other leads to further deviation from actual track surface. We believe this could be one of the reasons that controlling the sled while exiting a curve is difficult. Usually, the sled accelerates as the roll angle reduces and the cross-sections towards the end might have to be placed further apart.

6

Conclusions

We begin this chapter by summarising the work presented in this thesis and the results of this first attempt at developing a real-time simulator for the sport of Skeleton. We end by providing directions for future work that could help realize a real-time training simulator.

6.1. Conclusions

This work started out with the single goal of developing a real-time simulator for the sport of Skeleton. We haven't achieved that goal yet but we have come a very long way towards it. In Chapter 2, we presented a simple method to approximate the track surface used in the sport. This method relies on video footage and the use of Bezier curves. It's reliance on video footage is both a boon and a bane. Not being heavily reliant on construction data allows one to circumvent the lack of access to said data. The problem is that the approximated surface can deviate substantially from the actual surface due to human error. The repeated use of Bezier curve interpolation quite easily generates a surface that has been successfully used for simulations, albeit for partial descents.

In Chapter 3, we presented a model to describe the behaviour of the sled and athletes steering capabilities. It utilizes changes in distribution of contact forces between the two runners to allow for steering, this in line with previous literature [20, 23, 24]. In the simulated runs we also observe that one can steer the sled down the track using this model. In these simulated runs, of those that were performed at real-time speeds, the user is able to steer up to exit of curve 2 before encountering troubles. While for a simulation run four times slower, users could reach the fourth curve on a regular basis. Most of the troubles arise during the exit of curves. For these simulated runs, we used an integration time stepping scheme based on the Generalised-α method. This scheme proved to be quick enough for almost all time steps. This is a great sign for us as we did not put any efforts towards optimizing the run time of the MATLAB code.

We chose a video game controller to serve as input and visualize the position and orientation along the track on a stationary screen. With all the bits and pieces of the simulator setup and working, we attempted to validate the simulator comparing times for traversing a segment of the track and the sum of contact Lagrangian multipliers. This was met with mixed results. The elapsed time for simulated runs 15% slower than actual descents. While the Lagrangian multipliers showed a behaviour that was expected, closely following the weight of the sled acting normal the track surface. We are quite pleased with the performance of the simulator and hope that future improvements can bring the simulator to a stage where athletes use it. We present possible directions for future work in the section that follows

6.2. Recommendations for Future Work

We believe this simulator lays a solid foundation to explore multiple topics related to real-time simulation of the sport. The shear number of parameters mean that multiple studies can be done to determine their effects on the simulation. Take for example the magnitude of integration damping ρ_∞, it could be that it is introducing excessive damping and thus an optimal value other than 0, needs to be determined. The dynamic model

(Chapter 3) can also be extended to include the effect of contact between the athlete-sled body and the track surface other than the runner contact. This could mean extending the current system of equations with those that describe impact like in [6].

For a second direction on improvements, we go back to the original motivation for this project and ask, how might this be made into a 'training simulator' that athletes use on a regular basis?

We have previously noted that the track model created using the method presented in Chapter 2 is a rough approximation of the actual surface. The more accurately the surface can be represented, the more beneficial will the experience of using the simulator be for the athletes. To this end, a better method must be developed. A method that could eliminate human error from this process would be interesting direction to start with. Since track authorities are reluctant to grant access to construction data, this new method would also have to rely on using the video footage of descents, which is more easily obtainable.

Concurrent to the development of this simulator at TU Delft, there was a project to develop a sensor package to instrument a Skeleton sled. One part of this package are novel force sensors that measure the forces an athlete applies on the sled through their shoulders and knees. We believe using such an instrumented sled as the input device will drastically improve the realism and immersion for athletes using the simulator when compared to using the controller.

Continuing on the immersion a user feels as they use the simulator, Levy and Katz [7] highlight several ideas. They note sound and visual sensation of speed as important to improving immersion. In contradiction to other simulators developed for Bobsleighing [6, 8], they chose against using a motion platform to simulate roll rotation during a descent as coaches believed it would provide athlete with incorrect feedback. Realism and immersion aside, questions regarding feedback and analysis of simulated runs need to be answered. The use of the instrumented sled for practice runs on track and simulated runs will lead to much more data being made available to athletes and coaches. Making this data useful to them will be a challenge of its own.

A

Skeleton - The Sport

A.1. Introduction

Skeleton is one of three track-bound sliding sports that feature in the Winter Olympics, Bobsleigh (or Bobsled) and Luge being the other two. Skeleton was permanently added to the Winter Olympics programme in 2002. Bobsleigh and Skeleton are governed by a single body the International Bobsleigh and Skeleton Federation (IBSF).

The differences between these three sliding sports include the sled used, the ice skates (or runners), the number of participants and the posture maintained by the athletes during a run. In Skeleton, athletes ride on ice covered tracks in a head first, prone position [22]. Another key difference between the three sports, is at the start of each run as shown in Fig. A.1. With Skeleton, the athlete sprints while pushing the sled alongside them until they have reached their top speed and then 'load' by jumping on to the sled and assuming race position. This is a key component of the competition, as once they load onto the sled, gravity is the sole accelerating force. No means of propulsion are allowed, yet speeds in excess of 130 km/h are routinely achieved.

(a) Skeleton

(b) Luge

(c) Bobsleigh

Figure A.1: Start phases of track-bound winter sliding sports. Both Skeleton and Bobsleigh involve pushing a sled while running. In Luge start phase, athletes use upper limbs for propulsion.

The athlete steers by changing the distribution of his weight on the sled or by applying forces on the sled to

deform it and this in turn changes the contact area and force between the runners and track surface which changes the distribution of friction forces, resulting in lateral forces and a turning moment about the centre of mass and loss in energy [22, 23].

A.2. Skeleton Sled

From Roberts [24],
“The skeleton sled comprises of three main elements, (Fig 1.2):

1. Frame: the frame of the skeleton sled is a steel construction, approximately shoulder width and shoulder to knee in length. In its most basic form, it is a rectangular steel frame with a centre saddle to support the athlete's body.

2. Bellypan: mounted onto the underside of the frame is a smooth outer covering usually of fibreglass or carbon fibre composite which not only acts as an aerodynamic aid sealing the underside of the sled but also ensures the outer of the sled is smooth for athlete protection.

3. Runners: securely fixed to the left and right side of the frame are two steel runners that pass through the bellypan at the mounting points and run beneath the sled. They are the components in direct contact with the ice which enable the athlete to control the position of the sled."

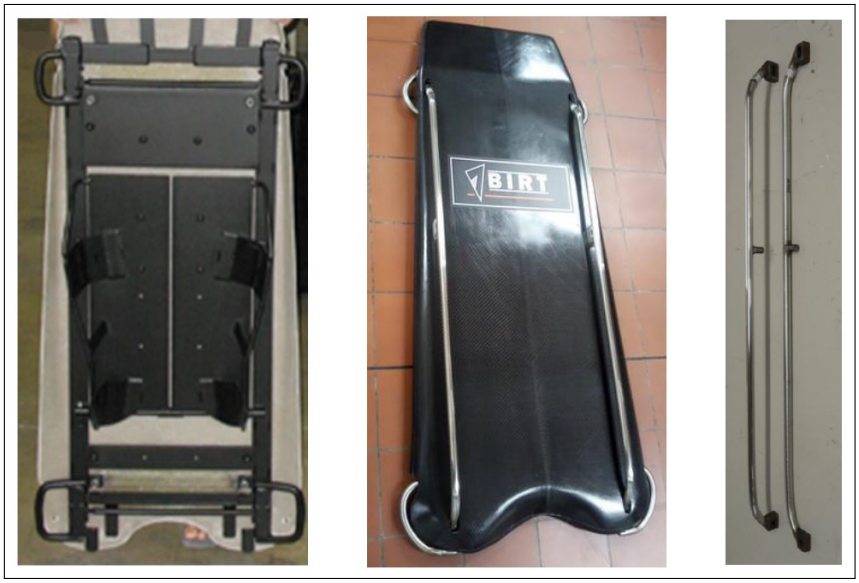


Figure A.2: The three main components of a Skeleton sled. From left to right, the frame, the bellypan and the runners. From [24]

A.3. Runner-Ice Surface Interaction

Apart from aerodynamic drag, the other resistive force to the sliding motion in these winter sports is the friction between the runners and ice surface. It is estimated that a change in coefficient of friction from 0.004 to 0.006 is sufficient to cause a potential first place finish to end up outside the top six [25]. In reality, it is not just a question of minimizing the coefficient of friction at all costs, but is actually a compromise between steering control and friction [24]. The lower the coefficient of friction, the lower the effect of steering actions as well. So with reduced friction, an athlete will slide faster but will have less control over the direction of sliding. There have been a few attempts to determine the coefficient of friction of Bobsleigh runners from experiments, the results are summarized in Table A.1.

Coeff. of Friction (x 10^{-3})	Velocity in m/s	Ice Temp. in ^oC	Source
9 ± 2.5	1.5	-1 to -10	[26]
4.2 ± 0.9	1 to 10	-2.2 to -4.6	[27]
5.3 ± 2.0	16 to 35	-10.6	[28]
4 ± 1	5	Not stated	[29]

Table A.1: Values of coefficient of friction for Bobsleigh runners sliding on ice determined from experiments

The parameters that have been observed to affect runner-ice friction include:

- Ice temperature [25]
- Air humidity [30]
- Shape of runners [31]
- Material properties of runner
  - Heat transfer coefficient [25]
  - Elastic strength [25]
- Sliding speed [25, 32]

Based on a model for speed skating blades Lozowski et al. developed a model for the friction between Bobsleigh runners and ice surfaces [31]. This model was later extended to Skeleton runners [20]. The model assumes that the friction force between runner and ice surface is the sum of two forces. The first force is due to the plastic deformation of the ice as the runner penetrates the ice surface and is called the ploughing force, F_p. A key aspect is the contact geometry. Bobsleigh and Skeleton runners have a longitudinal and a transverse radii and the resulting contact area is a semi-ellipse. The longitudinal radius is commonly referred to as the 'rocker' radius. Figure A.3 shows the key variables of this model when viewed normal to the direction of sliding.

The second force arises from the flow of the liquid water layer that is present on the ice surface and is the shear force, F_s. The thickness of this layer is increased by heat conduction into ice and by frictional heating between runner and ice. As the runner slides over the ice, this water is squeezed out by the pressure between runner and ice surface as seen in Fig. A.4. Due to the viscosity of water, a shear force acts opposite to the direction of motion.

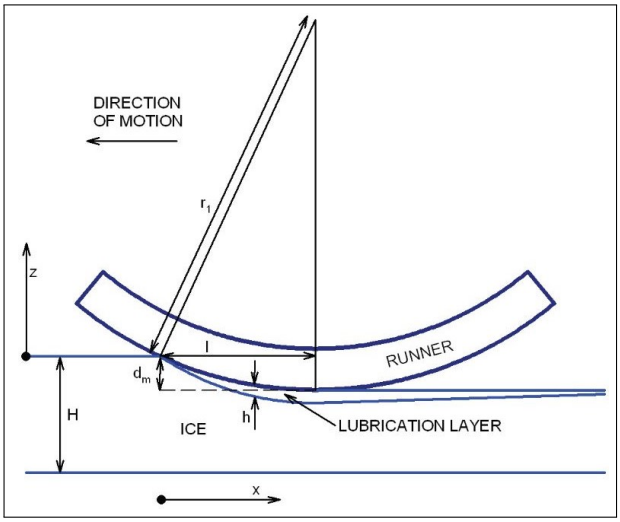


Figure A.3: The figure shows the deformation of ice surface as the runner slides along it, where longitudinal (rocker) radius is r_1, the maximum depth of penetration is d_m, thickness of lubrication layer is h and length of contact region is l. Taken from [31].

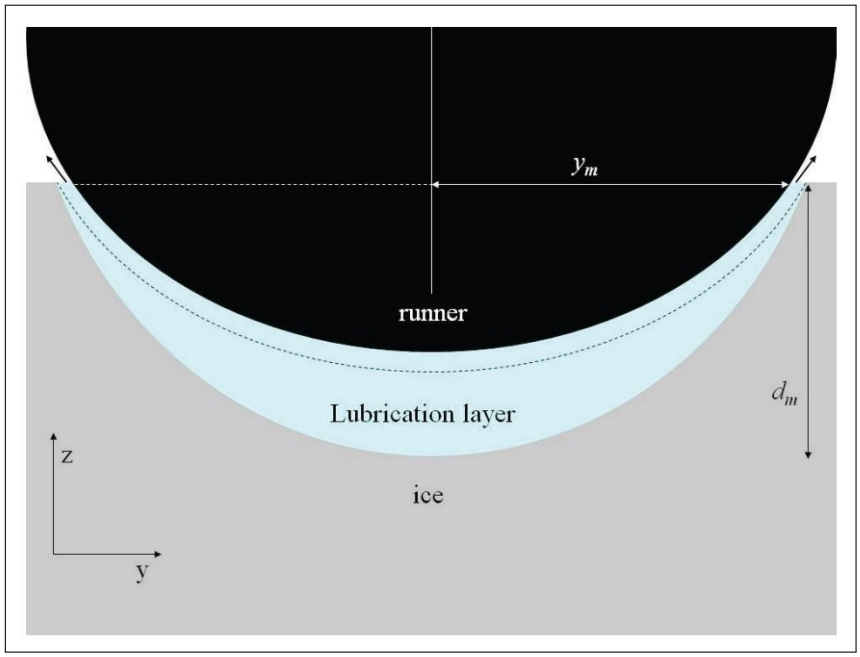
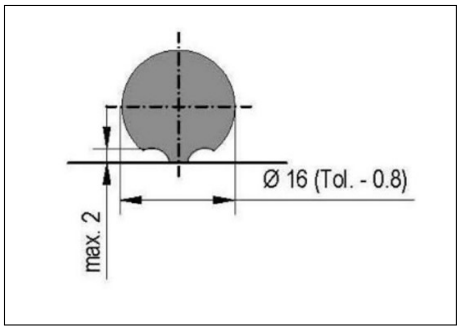


Figure A.4: The figure shows the transverse cross section of the runner at maximum depth of penetration while it slides across the ice surface, where the arrows outward from the lubrication layer at ice surface show the direction of flow of the water. d_m is maximum depth of penetration and y_m is half the contact width at that point. Taken from [31]

A.4. Steering In Skeleton

Compared to steering in Bobsleigh where the front pair of runners can be rotated, steering in Skeleton is a more complex process primarily involving changes in the runner-ice contact geometry and force distribution. A Skeleton runner differs from a Bobsleigh runner due to presence of a spine which is the result of one or two grooves that are cut in to the runners. These grooves are what allow for steering manoeuvres to be possible from the deformation of the runners and also provide increased lateral stability when the runners penetrate deeper into the ice surface. Figures A.5a and A.5b show an example of the runners and the restrictions placed by the IBSF on the dimensions of the runner and grooves.

(a) Grooves with resulting spine. From [24]



(b) IBSF restrictions on grooves. Dimensions in mm. From [18]

Figure A.5: Grooves found on the rear half a Skeleton runner

There are 3 methods of steering in Skeleton categorized based on body part used and they are[1] [20, 23, 24, 33]:

1. Head steering
2. Shoulder and Knee steering
3. Toe steering

[1] Personal Correspondence - Akwasi Frimpong, Skeleton Athlete

Head Steering

This method of steering is used for small direction changes in straight sections. The athletes tilt their head towards the side they wish to turn as shown in Fig. A.6. This leads to a small change in the distribution of aerodynamic drag forces and the weight distribution between the two runners. The magnitude of steering achieved by this method is small and it is restricted to straight sections of the track where shoulder and knee steering cannot be used effectively.

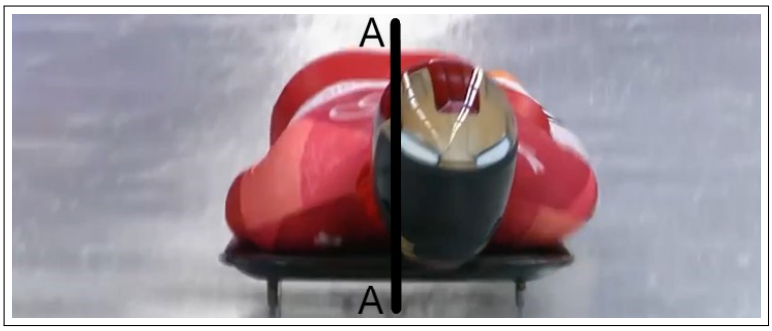


Figure A.6: A Skeleton athlete attempting to steering using his head, the unequal distribution of body mass about a vertical plane at the mid-point of sled width approximated by A-A can be seen.

Shoulder and Knee Steering

This method of steering involves applying forces with the shoulders and knees on to the sled frame, deforming it. This deforms the runners, changing the contact area and increasing the friction force on one runner compared to the other. Athletes use either a single shoulder or a single knee, or they use a combination of a shoulder and a knee to apply the force as shown in Fig. A.7. When a shoulder is used, it shifts the contact area forward compared to the no force applied condition. This reduces the length of spine contact of that runner. Subsequently friction on the side of the shoulder used is reduced and the sled turns towards the opposite side. When a knee is used, the opposite is true. Contact area shifts backward, length of spine contact increases, friction on that runner increases and sled turns to the side of application of force. Changes in contact area are visualized in Figure A.9. A combination of shoulder and opposite side knee are used when large steering forces need to be applied.

In his thesis, I. Roberts describes two types of sleds based on the steel frame construction, they are shown in Figure A.8. While not highlighted in the figure, sled B has a more rigid frame compared to sled A. The pivot point of Sled A is closer to the shoulders than the knee. This according to Roberts leads to knee-dominant steering, possibly as the bending moment acting on the hinge increases due to increase in distance from hinge to knee. While sled B with its pivot point almost half way between shoulder and knee does not favour a particular limb of steering like sled A does [24].

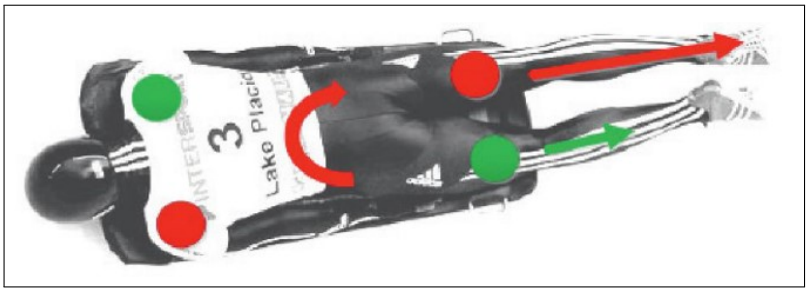


Figure A.7: By applying forces with his left shoulder and/or his right knee (represented by red dots) the athlete deforms the sled and the runners such that friction force on the right side runner is increased compared to the left side runner causing the sled to turn about the yaw axis to the right as shown by the red arrow. From [23]

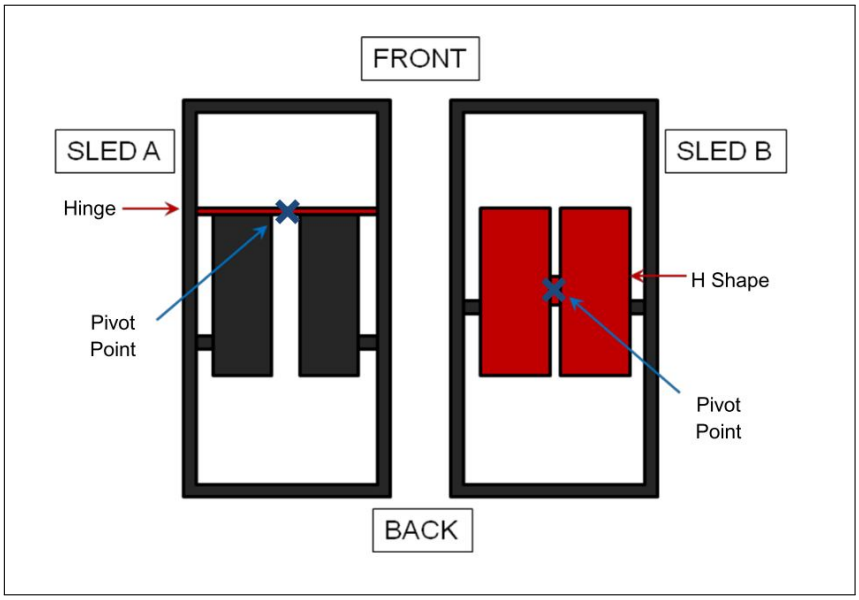
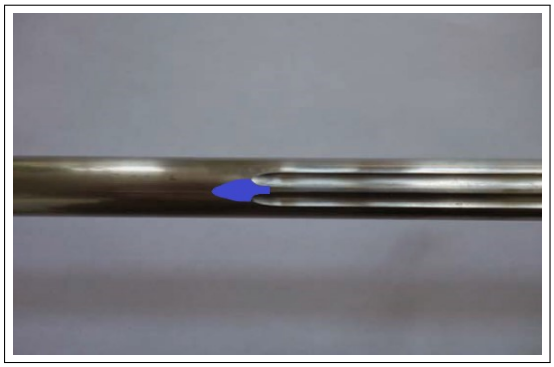


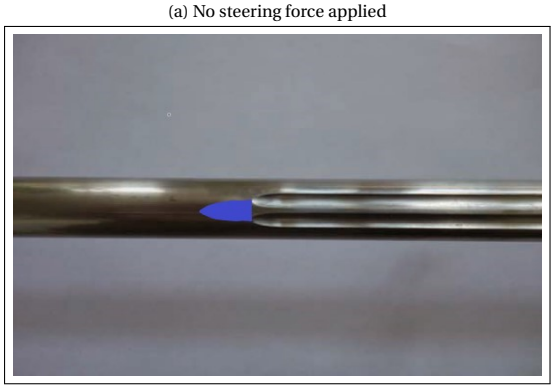
Figure A.8: Representation of two types of Skeleton sled frame constructions from [24]. Based on the position of the hinge and the pivot point, sled A favours knee steering over shoulder steering, while sled B does not favour one over the other.

Toe steering

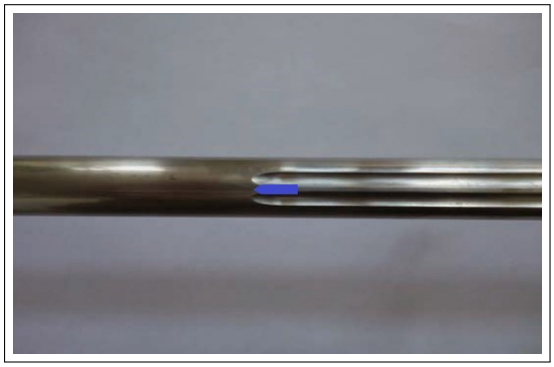
During a run, the limbs of an athlete normally do not make contact with the track surface. Toe steering involves striking the ice surface with the toes and is also known as Toe tapping. This adds an additional contact point with the surface and leads to a turning moment about the centre of gravity. Use of toes is considered an emergency steering action as it leads to a substantially larger reduction of speed and is more commonly used by beginners.



(a) No steering force applied



(b) Steering force applied by shoulder



(c) Steering force applied by knee

Figure A.9: Changes in contact area of skeleton runner when shoulder or knee is used to deform sled and runner according to Lozowski et al. [20]. The area shaded in blue is the region of the runner that penetrates into the ice surface. When the shoulder is used the contact area moves forward and spine contact length is reduced as can been seen by comparing A.9a to A.9b. The opposite happens when the knee is used as shown in A.9c.

B

Bezier Curves

B.1. Introduction

Bezier Curves were chosen for use in two interpolation steps, for the approximation of individual transverse cross-sections of curved segments and for interpolation between the transverse cross-sections. This chapter briefly explains the characteristics of Bezier curves.

Bezier curves are a type of parametric polynomial curve [13]. They are a special case of B-spline which does not possess any knots other than the first and last control points. A Bezier curve $\boldsymbol{C}(u)$ is defined as,

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} B_{i,n}(u)\ \boldsymbol{p}_i \qquad\qquad 0 \le u \le 1 \tag{B.1}$$

where, n is the degree of the Bezier curve defined by $(n+1)$ control points $\boldsymbol{p}_i$. In the case of approximating individual transverse cross-sections, the points obtained from the video frame are the control points. And, $B_{i,n}(u)$ are the Bernstein polynomials and are defined as,

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!}\ u^i\ (1-u)^{n-i} \tag{B.2}$$

The Bernstein polynomials can be recursively defined as,

$$B_{i,n}(u) = (1-u)\ B_{i,n-1}(u) + u\ B_{i-1,n-1}(u) \tag{B.3}$$

that leads to a second definition for Bezier curves, representing them as the result of repeated linear interpolations. For example, a quadratic Bezier curve ($n = 2$) can be computed from interpolation between two linear Bezier curves as shown below,

$$\begin{aligned}\boldsymbol{C}(u) &= \sum_{i=0}^{2} B_{i,2}(u)\ \boldsymbol{p}_i \\ &= (1-u)^2\ \boldsymbol{p}_0 + 2u(1-u)\boldsymbol{p}_1 + u^2\boldsymbol{p}_2 \\ &= (1-u)((1-u)\boldsymbol{p}_0 + u\boldsymbol{p}_1) + u((1-u)\boldsymbol{p}_1 + u\boldsymbol{p}_2)\end{aligned}$$

where, $(1-u)\boldsymbol{p}_0 + u\boldsymbol{p}_1$ and $(1-u)\boldsymbol{p}_1 + u\boldsymbol{p}_2$ are two linear Bezier curves that are straight lines from $\boldsymbol{p}_0$ to $\boldsymbol{p}_1$ and $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ respectively.

Figure B.1 shows the process of repeated linear interpolation for a cubic Bezier curve (n=3) with 4 control points and for $u = \frac{2}{5}$. The points $\boldsymbol{p}_{1,0}, \boldsymbol{p}_{1,1}, \boldsymbol{p}_{1,2}$ are obtained by interpolation between points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ respectively. Points $\boldsymbol{p}_{2,0}, \boldsymbol{p}_{2,1}$ are obtained by interpolation between $\boldsymbol{p}_{1,0}$ and $\boldsymbol{p}_{1,1}$, $\boldsymbol{p}_{1,1}$ and $\boldsymbol{p}_{1,2}$ respectively. Finally point $\boldsymbol{p}_{3,0}$ is obtained by interpolation between $\boldsymbol{p}_{2,0}$ and $\boldsymbol{p}_{2,1}$. For all the linear interpolation operations the value of the parameter u is $\frac{2}{5}$. By varying u from 0 to 1, one obtains the complete

cubic Bezier curve from $\boldsymbol{p}_0$ to $\boldsymbol{p}_3$ defined by control points $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$.

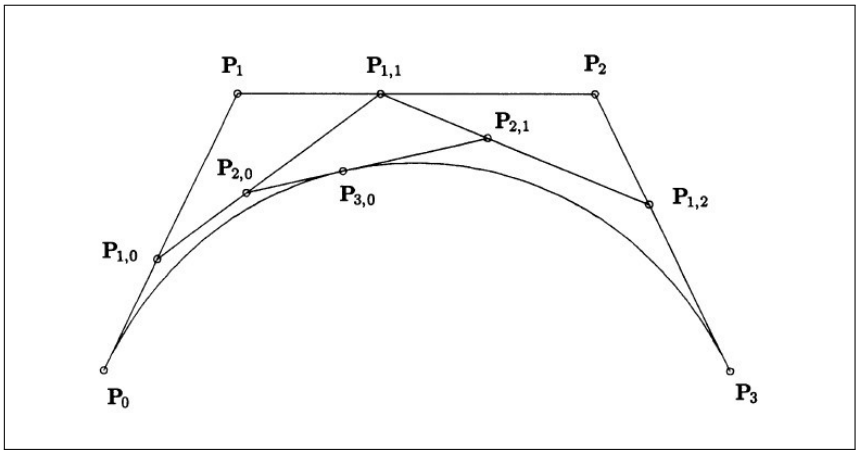


Figure B.1: Cubic Bezier Curve obtained through repeated linear interpolation. From [13]

B.2. Derivatives

B.2.1. Definitions

The first derivative of a Bezier curve is,

$$\begin{aligned}\boldsymbol{C}^1 &= \frac{d}{du}\left(\sum_{i=0}^{n} B_{i,n}(u)\,\boldsymbol{p}_i\right)\\ &= \sum_{i=0}^{n} B^1_{i,n}(u)\,\boldsymbol{p}_i\\ &= n\sum_{i=0}^{n-1} B_{i,n-1}(u)\,(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)\end{aligned} \tag{B.4}$$

At the end points, $u = 0$ and $u = 1$, the derivatives reduce to,

$$\boldsymbol{C}^1(0) = n(\boldsymbol{p}_1 - \boldsymbol{p}_0) \tag{B.5}$$

$$\boldsymbol{C}^1(1) = n(\boldsymbol{p}_n - \boldsymbol{p}_{n-1}) \tag{B.6}$$

Similarly the second derivative is,

$$\begin{aligned}\boldsymbol{C}^2 &= \frac{d}{du}\left(n\sum_{i=0}^{n-1} B_{i,n-1}(u)\,(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)\right)\\ &= n\sum_{i=0}^{n-1} B^1_{i,n-1}(u)\,(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)\\ &= n(n-1)\sum_{i=0}^{n-2} B_{i,n-2}(u)\,(\boldsymbol{p}_{i+2}-2\boldsymbol{p}_{i+1}+\boldsymbol{p}_i)\end{aligned} \tag{B.7}$$

Comparing equations B.4 & B.7 with B.1, it is clear that the k^{th} derivative of a Bezier curve of degree n is a Bezier curve of degree $(n-k)$. Hence $(n-1)$ non-zero derivatives exist for such a curve.

The second derivatives at the end points $u = 0$ and $u = 1$ are given by,

$$\boldsymbol{C}^2(0) = n(n-1)(\boldsymbol{p}_0 - 2\boldsymbol{p}_1 + \boldsymbol{p}_2) \tag{B.8}$$

$$\boldsymbol{C}^2(1) = n(n-1)(\boldsymbol{p}_n - 2\boldsymbol{p}_{n-1} + \boldsymbol{p}_{n-2}) \tag{B.9}$$

B.2.2. Continuity

As a type of B-splines, Bezier curves are also infinitely differentiable in the span between two knots. Since a Bezier curve does not posses any knots other than its end-points, such a curve will also be infinitely differentiable and hence infinitely continuous except for at its end-points. Thus when using Bezier curves to describe the track surface, we have to consider continuity at the transition between segments.

Let $\boldsymbol{C}_1(u)$ be a Bezier curve of degree n defined by $(n+1)$ control points and $\boldsymbol{C}_2(u)$ be an adjoining Bezier curve of degree m defined by $(m+1)$ control points, such that $\boldsymbol{p}_n$ of curve $\boldsymbol{C}_1(u)$ is $\boldsymbol{p}_0$ for curve $\boldsymbol{C}_2(u)$ or in other words,

$$\boldsymbol{C}_1(1) = \boldsymbol{C}_2(0)$$

For such curves, there are two types of continuity, a geometric continuity G^k and parametric continuity C^k. For visual purposes often G^2 continuity is sought after as it means that the curvature vector is continuous and the human eye is not good at distinguishing discontinuities of third-order or higher derivatives. For physics simulation we would like to have curves or surfaces with C^2 continuity, which means the acceleration vector is continuous. To obtain this continuity when transitioning from curve $\boldsymbol{C}_1$ to $\boldsymbol{C}_2$, the second derivatives should be equal at the common point. That is $\boldsymbol{C}_1^2(1) = \boldsymbol{C}_2^2(0)$ and from Eqn. (B.8) and (B.9),

$$n(n-1)(\boldsymbol{p}_n - 2\boldsymbol{p}_{n-1} + \boldsymbol{p}_{n-2}) = m(m-1)(\boldsymbol{p}_0 - 2\boldsymbol{p}_1 + \boldsymbol{p}_2) \tag{B.10}$$

Let the $(n+1)$ control points of curve $\boldsymbol{C}_1$ and $(m+1)$ control points of curve $\boldsymbol{C}_2$ be represented by a common index o where, $o = 0,1,2,3,...,(n+m+1)$. Then for C^2 continuity at the common control point $o = (n+1)$, equation B.10 is,

$$n(n-1)(\boldsymbol{p}_o - 2\boldsymbol{p}_{o-1} + \boldsymbol{p}_{o-2}) = m(m-1)(\boldsymbol{p}_o - 2\boldsymbol{p}_{o+1} + \boldsymbol{p}_{o+2}) \tag{B.11}$$

Therefore the two conditions for C^2 parametric continuity at the junction between two Bezier curves are:

1. $n = m$
2. points $\boldsymbol{p}_{o-2}, \boldsymbol{p}_{o-1}, \boldsymbol{p}_o, \boldsymbol{p}_{o+1}, \boldsymbol{p}_{o+2}$ are collinear and the distance between pairs of successive points are equal.

If only condition 2 above is satisfied, that is

$$(\boldsymbol{p}_o - 2\boldsymbol{p}_{o-1} + \boldsymbol{p}_{o-2}) = (\boldsymbol{p}_o - 2\boldsymbol{p}_{o+1} + \boldsymbol{p}_{o+2}) \tag{B.12}$$

we obtain G^2 geometric continuity.

B.3. Advantages

Along with the simplicity derived from the repeated interpolation definition, the properties of Bezier curves that favoured their use are:

L1.1 While the curve passes through the end points $\boldsymbol{p}_0$ and $\boldsymbol{p}_n$, it does not necessarily pass through the remaining control points. This is known as end-point interpolation property.

L1.2 Apart from the end points, Bezier curves have infinite continuity. At the endpoints $\boldsymbol{p}_0$ and $\boldsymbol{p}_n$, the continuity, C^k depends solely on the $k+1$ control points and the degree n at the respective ends. This property extends to a surface $S(u,v)$ defined by the tensor product of a bidirectional set of control points n and m. Where apart from the knot points, the surface will be infinitely continuous.

L1.3 Bezier Curves lack local control, each control point effects the entire curve but the effect diminishes as you move further away from that control point. For example in Figure B.1, changing any of the points $\boldsymbol{p}_i$ will affect the entire curve.

Bibliography

[1] N. Bullock, W. G. Hopkins, D. T. Martin, and F. E. Marino, "Characteristics of Performance in Skeleton World Cup Races," *Journal of Sports Sciences*, vol. 27, no. 4, pp. 367–372, 2009.

[2] N. Bullock and W. G. Hopkins, "Methods for Tracking Athletes' Competitive Performance in Skeleton," *Journal of Sports Sciences*, vol. 27, no. 9, pp. 937–940, 2009.

[3] R. M. Malcata and W. G. Hopkins, "Variability of Competitive Performance of Elite Athletes: A Systematic Review," *Sports Medicine*, vol. 44, no. 12, pp. 1763–1774, dec 2014. [Online]. Available: http://link.springer.com/10.1007/s40279-014-0239-x

[4] T. J. Mosey and M. R McGuigan, "The Relationship Between Competitive Experience and Performance in Skeleton," *Journal of Athletic Enhancement*, vol. 03, no. 03, pp. 1–4, 2014. [Online]. Available: http://www.scitechnol.com/the-relationship-between-competitive-experience-and-performance-in-skeleton-motD.php?article{_}id=1983

[5] J. Roche, S. Turnock, and S. Wright, "An Analysis of the Interaction Between Slider Physique and Descent Time for the Bob Skeleton," *The Engineering of Sport 7*, pp. 101–109, 2008.

[6] G. S. Rempfler and C. Glocker, "A Bobsleigh Simulator Software," *Multibody System Dynamics*, vol. 36, no. 3, pp. 257–278, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11044-015-9450-2

[7] R. M. Levy and L. Katz, "Virtual reality simulation: Bobsled and Luge," *IACSS International Symposium Computer Science in Sport*, pp. 1–11, 2007.

[8] A. Kelly and M. Hubbard, "Design and Construction of a Bobsled Driver Training Simulator," *Sports Engineering*, vol. 3, no. 1, pp. 13–24, 2000. [Online]. Available: http://www.blackwell-synergy.com/links/doi/10.1046/j.1460-2687.2000.00037.x

[9] F. Braghin, F. Cheli, M. Donzelli, S. Melzi, and E. Sabbioni, "Multi-body Model of a Bobsleigh: Comparison with Experimental Data," *Multibody System Dynamics*, vol. 25, no. 2, pp. 185–201, 2011.

[10] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Design and Verification of Bobsleigh Track," *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, ESDA2010*, vol. 4, pp. 505–512, 2010.

[11] G. Rempfler, "Development of a Bobsleigh Simulator (Entwicklung eines Bobsimulators)," Ph.D. dissertation, ETH, Zurich, 2015.

[12] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis," *Nature methods*, vol. 9, no. 7, p. 671, 2012.

[13] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.

[14] M. Hubbard, M. Kallay, and P. Rowhani, "Three-Dimensional Bobsled Turning Dynamics," *International Journal of Sport Biomechanics*, vol. 5, no. 1973, pp. 222–237, 1989.

[15] MATLAB, *version 9.6.0 (R2019a)*. Natick, Massachusetts: The MathWorks Inc., 2019.

[16] H. Vallery and A. L. Schwab, *Advanced Dynamics*. Delft University of Technology, 2017.

[17] J. Chung and G. M. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method," *Journal of Applied Mechanics, Transactions ASME*, vol. 60, no. 2, pp. 371–375, 1993.

[18] International Bobsleigh and Skeleton Federation, "International Skeleton Rules 2018," p. 44, 2018. [Online]. Available: http://www.ibsf.org/images/documents/downloads/Rules/2018{_}2019/2018{_}International{_}Rules{_}SKELETON{_}am.pdf

[19] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[20] E. Lozowski, K. Szilder, S. Maw, and A. Morris, "A Model of Ice Friction for Skeleton Sled Runners," in *Twenty-Fourth International Ocean and Polar Engineering Conference*, 2014.

[21] J. J. de Koning, G. De Groot, and G. J. van Ingen Schenau, "Ice friction during speed skating," *Journal of Biomechanics*, vol. 25, no. 6, pp. 565–571, 1992.

[22] R. Larman, S. Turnock, and J. Hart, "Mechanics of the Bob Skeleton and Analysis of the Variation in Performance at the St Moritz World Championship of 2007," in *The Engineering of Sport 7 - Vol. 2*. Paris: Springer Paris, 2008, pp. 117–125. [Online]. Available: http://link.springer.com/10.1007/978-2-287-09413-2

[23] C. Sawade, S. Turnock, A. Forrester, and M. Toward, "Assessment of an Empirical Bob-Skeleton Steering Model," *Procedia Engineering*, vol. 72, pp. 447–452, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.proeng.2014.06.078

[24] I. J. M. Roberts, "Skeleton Bobsleigh Mechanics: Athlete-Sled Interaction," Ph.D. dissertation, Univeristy of Edinburgh, 2013. [Online]. Available: http://www.era.lib.ed.ac.uk/handle/1842/7600{%}5Cnhttps://www.era.lib.ed.ac.uk/bitstream/1842/7600/1/Roberts2013.pdf

[25] P. Dabnichki, "Bobsleigh Performance Characteristics for Winning Design," *Procedia Engineering*, vol. 112, pp. 436–442, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.proeng.2015.07.221

[26] K. Itagaki, G. E. Lemieux, and N. P. Huber, "Preliminary Study of Friction Between Ice and Sled Runners." *Journal de Physique (Paris), Colloque*, vol. 48, pp. 297–301, 1987.

[27] L. Poirier, E. P. Lozowski, S. Maw, D. J. Stefanyshyn, and R. I. Thompson, "Experimental Analysis of Ice Friction in the Sport of Bobsleigh," *Sports Engineering*, vol. 14, no. 2-4, pp. 67–72, 2011.

[28] L. Poirier and R. I. Thompson, "Getting a Grip on Ice Friction," *International Offshore and Polar Engineering Conference*, vol. 8, pp. 1071–1077, 2011. [Online]. Available: http://e-book.lib.sjtu.edu.cn/isope2011/data/papers/11TPC-901Poirie.pdf

[29] M. Irbe, K. A. Gross, J. Viba, and M. Cerpinska, "Analysis of Acceleration and Numerical Modelling of Skeleton Sled Motion," *Engineering for Rural Development*, vol. 17, pp. 1401–1406, 2018.

[30] L. Poirier, E. P. Lozowski, and R. I. Thompson, "Ice Hardness in Winter Sports," *Cold Regions Science and Technology*, vol. 67, no. 3, pp. 129–134, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.coldregions.2011.02.005

[31] E. P. Lozowski, K. Szilder, and L. Poirier, "A Bobsleigh Ice Friction Model," *International Journal of Offshore and Polar Engineering*, vol. 24, no. 1, pp. 1259–1267, 2014. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84883697973{&}partnerID=40{&}md5=99c2f74e6a8d18d646d46b27f63c4c4c

[32] M. Scherge, R. Böttcher, A. Spagni, and D. Marchetto, "High-Speed Measurements of Steel–Ice Friction: Experiment vs. Calculation," *Lubricants*, vol. 6, no. 1, p. 26, 2018.

[33] C. Gong, C. W. G. Phillips, E. Rogers, and S. Turnock, "Analysis of Performance Indices for Simulated Bob-Skeleton Descents," in *11th Conference of the International Sports Engineering Association, ISEA 2016*, vol. 00, 2016, pp. 1–6.