

Diss. ETH Nr. 22546

Entwicklung eines Bobsimulators

Abhandlung zur Erlangung des Titels

DOKTOR DER WISSENSCHAFTEN der ETH ZÜRICH

(Dr. sc. ETH Zürich)

vorgelegt von

GEORG STEFAN REMPFLER

MSc ETH Masch.-Ing.

geboren am 13. November 1984

von

Appenzell, Appenzell Innerrhoden

Schweiz

angenommen auf Antrag von

Prof. Dr.-Ing. Dr.-Ing. habil. Ch. Glocker, Referent

Prof. Dr. R. Riener, Korreferent

2015

Zusammenfassung

Die vorliegende Dissertation beschreibt die zentralen Elemente eines Bobsimulators, der im Rahmen dieses Projektes entwickelt, aufgebaut und getestet wurde. Er dient Profis und Amateuren in erster Linie als Trockentrainingsgelegenheit, eignet sich aber auch für die Analyse von Bahndesigns oder Konstruktionsvarianten von Schlitten. Die Simulation einer gesteuerten Bobfahrt ist eine prädestinierte Anwendung für die Methoden der nicht glatten Dynamik. Ihre Formulierung von mengenwertigen Kraftgesetzen in Form von Normalkegelinklusionen eignet sich insbesondere für die Modellierung des einseitigen, reibungsbehafteten Kontaktes zwischen Kufe und Eis. Mit der Formulierung von Reibstößen bei denen impulsive Kräfte zu Sprüngen in den Geschwindigkeiten führen, ist das temporäre Abheben der Kufen vom Eis im Modell enthalten und sind auch Schläge gegen die seitliche Berandung der Bahn mitberücksichtigt. Mit der iterativen Lösung des Inklusionsproblems, das sich in einem Zeitschritt für die Kontaktkräfte stellt, und den Time-Stepping Verfahren zur numerischen Integration sind robuste Instrumente für die Simulation nicht glatter Modelle vorhanden. Aufgrund der im Bobsport üblichen hohen Geschwindigkeiten von bis zu 150km/h stellt die Simulation einer Fahrt einen Prüfstein für die Effizienz der Methoden dar. Denn die Simulation hat in Echtzeit zu erfolgen und muss dabei stets flüssig auf der Leinwand dargestellt werden können.

Für den Bobschlitten mitsamt Mannschaft wird ein Mehrkörpermodell aus Starrkörpern entwickelt, das die wesentlichen Freiheitsgrade beinhaltet. Hier kann vom Know-How des Projektes *Citius* profitiert werden. Einzelne Krafterelemente im Schlitten sind der effektiven Bauart oder Messungen der modernen Schlitten nachempfunden. Für die anisotrope Reibung zwischen den Kufen und dem Eis wird ein eigenes Modell erarbeitet, das verschiedene Aspekte der Interaktion berücksichtigt. Insbesondere ist die mangelnde Lenkbarkeit eines Schlittens unter fehlender Last nachempfunden. Weiter kann der Schlitten von der Eisoberfläche abheben und damit auch stürzen. Er stösst aber nicht nur mit den Kufen gegen das Eis, sondern kann mit den seitlichen Bumpen auf Geraden an die Eiskanalanwand oder in Kurven an die Bretterschläge stossen. Solche Schläge werden häufig als Fehler taxiert, können aber auch einen driftenden Schlitten wieder ausrichten und so trotz Geschwindigkeitsverlust nützlich sein. Das Mehrkörpermodell interagiert mit einem Modell der Bahnoberfläche. Anhand von Konstruktionsdaten wird die Betonoberfläche von Kunsteisbahnen rekonstruiert. Aus Bildmaterial und Videos werden zusätzliche Daten aufgenommen für das Hinzufügen von Wänden, Bretterschlägen und Absperrungen von weiteren Starteinläufen in die Bahn. Das Bahnmodell wird in der dreidimensionalen Szenerie mit weiteren umgebenden Details ausgeschmückt. So können Schattenwürfe, Lampen

und Sonnenblenden verschiedene, reale Fahrsituationen nachahmen. Mit Texturen oder der Platzierung von Objekten können auch Orientierungspunkte der Athleten eingebaut werden.

Der Simulationscode nutzt die Baumstruktur des Mehrkörpermodells aus und stellt die Bewegungsgleichungen für einen Zeitschritt numerisch auf. Als Einführung in die Programmierung von Mehrkörpersimulationen werden in dieser Arbeit die wesentlichen Objekte der Softwarearchitektur beschrieben und deren Interaktion anhand von Python-Code gezeigt. Die vorgestellte Methode eignet sich speziell für den stufenweisen Auf- oder Ausbau eines Mehrkörpermodells, in dem wiederholt neue Körper, Freiheitsgrade oder Krafftelemente eingebaut werden. Grundsätzlich werden im Code die Bewegungsgleichungen in Minimalgeschwindigkeiten formuliert, weshalb neben den einseitigen Kontakten keine weiteren Bindungen berücksichtigt werden müssen. Gleichzeitig geschieht die Modellierung des Mehrkörpersystems im dreidimensionalen Anschauungsraum, was den Prozess sehr intuitiv macht.

Das User Interface des Bobsimulators ist bewusst minimal und intuitiv gehalten. Es erlaubt den Bobpiloten den Simulator ohne Anleitung zu nutzen und selbstständig zu trainieren. Über eine Eingabemaske lassen sich verschiedene Bobsetups individuell zusammenstellen, speichern und laden. Dabei werden einige Einstellungen zur Auswahl vorgegeben, die verschiedene real existierende Federsätze widerspiegeln. Der Simulator bietet neben einer (foto-)realistischen Darstellung der Szenerie auch Audio- und haptisches Feedback.

Abstract

This thesis describes the core elements of the bobsleigh simulator, that has been developed, built and tested within this project. It serves as an all year training opportunity for professional pilots and amateurs. Moreover, it leverages the analysis of track and bobsleigh designs. The simulation of a bobsleigh is a predestined application for the methods of non-smooth dynamics. Its formulation of set-valued force laws as normal cone inclusions is especially well suited for the modelling of the unilateral, frictional contact between a slider and the ice. With the formulation of frictional impacts, at which impulsive forces lead to a discontinuity in the velocities, the sliders may take-off from the ice and bumper hits against the side walls of the track are incorporated in the model too. With the iterative solver for the inclusion problem, that is stated for the contact forces in each time step, and the time-stepping schemes for the numerical integration of the motion, robust tools are given for the simulation of non-smooth models. Due to the high velocities of up to 150km/h occurring in the sport of bobsleighbing, the simulation of a bobsleigh marks a benchmark problem for the efficiency of these methods. The simulation has to be performed in real-time and must be smoothly displayed on a screen.

A multi-body model with the essential degrees of freedom consisting of rigid bodies is developed for the bobsleigh including the two-men team. Here, it was made use of the knowledge from the former project *Citius*. Some force elements in the model mimic the construction of the modern *Citius* bobsleighs, or measurements taken of them. For the anisotropic friction between the sliders and the ice a new model is introduced, which reflects different aspects of the interaction. Especially, the reduced controllability of a bobsleigh under low load is replicated. Moreover, the bobsleigh can lift off the ice and therefore may turn-over. However, it does not only impact against the environment over the sliders. The side bumpers can collide with the walls of the track on straight sections or hit the wooden planking in curves. Such bumps are often considered a driving mistake, but they may also bring a drifting bobsleigh back into straight direction and can therefore be helpful too, despite the accompanying loss in velocity. The multi-body model interacts with a model of the track surface, which is reconstructed from the construction data of artificial ice tracks. Additional data is gathered from images and videos for to add side walls, plankings in curves and barriers at alternative starts further down the track. The model of the track is accompanied with further details in the surrounding three-dimensional scenery. Different realistic driving situations can be rendered. Shadows and therewith transitions to enlightened areas may disturb a pilot. Artificial light from lamps creates an entirely different situation and closed sun shades tunnel the view on the track. With textures or

by placing additional objects reference points of the athletes can be included in the scene.

The simulation code makes use of the tree structure of the multi-body model and numerically assembles the equations of motion in each time step. As an introduction to the programming of multi-body simulations the core objects of the software architecture are described and their interaction is presented in Python code. The method used herein is well suited for a step by step build-up or extension of a multi-body model, where new bodies, degrees of freedom or force elements are continuously added. Basically, the code formulates the equations of motion in minimal velocities. Therefore, beside the unilateral contacts, no additional constraints have to be taken into account. On the other side, the modelling of the multi-body system happens in the three-dimensional, euclidean space, which makes this process very intuitive.

The user interface of the bobsleigh simulator is intentionally kept minimal and intuitive. It enables the pilots to use the simulator without instruction, and to self-dependently exercise on it. Over an input mask the bobsetup can individually be adjusted, saved and loaded. Some options can alternatively be chosen from predefined lists, that represent the physically existing spring sets given to *Citius* pilots. Aside the (photo-)realistic rendering of the scenery, the bobsleigh simulator provides audio and haptic feedback.