Zurich University of Applied Sciences

Department School of Engineering

InIT, ICP, IAMP

PE FS25

Semesterarbeit Teil 2 (Rollbahn mit Druckfedern)

Authors:
Michael Voemel
Leonard Bödi

Instructor:
Dr. Kurt Pernstich

Submitted on
April 7, 2025

Study program:
Computer Science, BSc

Contents

Chapter 1

Compression Springs

In all tasks we used the values vinit = 1, kF = 10 and dt = 20ms. Left bumper is set to 1kg.

The following chapter is the solution to the task ("1. Druckfedern").

1.1 The initial situation

In this scenario, the car (mass = 400g) moves between two bumpers. We simulate two virtual compression springs with the following parameters:

- Rest length: 15 cm
- Spring constant: 10 N/m

The springs are not rendered in the scene but exert forces via AddForce() whenever they are compressed. Compression only occurs if the distance between the car and the bumper is less than the spring's rest length.

1.2 Implementation Overview

```
// Calculate compression of the spring
compressionLeft = springLength - distanceToLeftBumper;
// Calculate force (F = k * x)
forceLeft = springConstant * compressionLeft;
// Apply force to the car
rb.AddForce(new Vector3(0, 0, forceLeft), ForceMode.Force);
```

1.2.1 Explanation

This logic is mirrored for the right bumper. When both springs are implemented, the car oscillates between the bumpers due to alternating forces from the springs.

For each spring, the code:

1. Calculates the distance between the car and the bumper
2. Determines if the spring is compressed (distance < rest length)

3. Calculates the compression amount and resulting force using Hooke's law ($F = k * x$)

4. Applies the force to the car and, if applicable, to the left bumper

This causes the car to oscillate between the bumpers, with the springs pushing it away when compressed.

1.2.2 Plots and Interpretation

We analyzed the car's position, velocity, and acceleration over time. The plots clearly show periodic oscillations. Acceleration spikes occur during spring compression, while velocity reverses direction at each peak compression point.

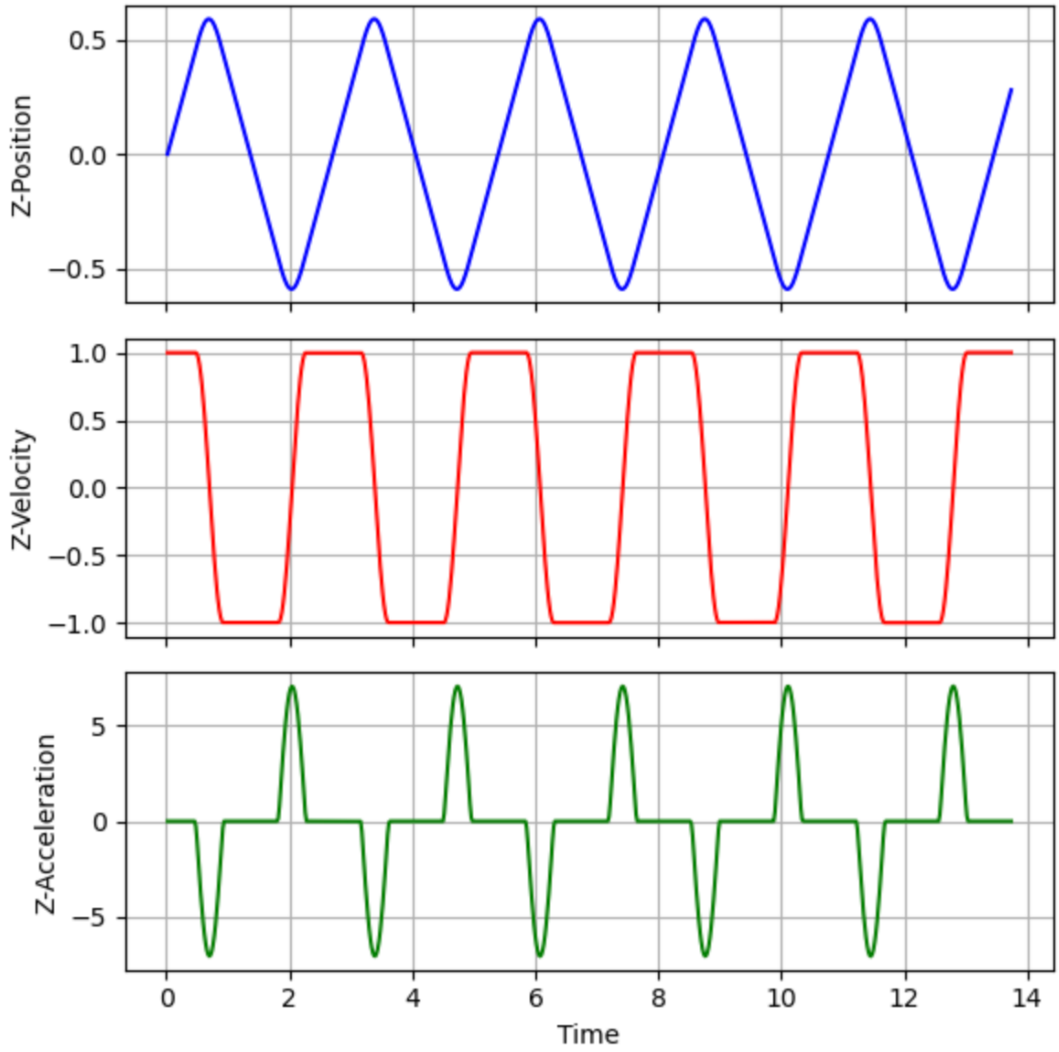


FIGURE 1.1: Plots showing the car's velocity with bumper mode 0

Chapter 2

Collision Formulas

The following chapter is the solution to the task ("2. Stossformeln").

2.1 Initial Situation

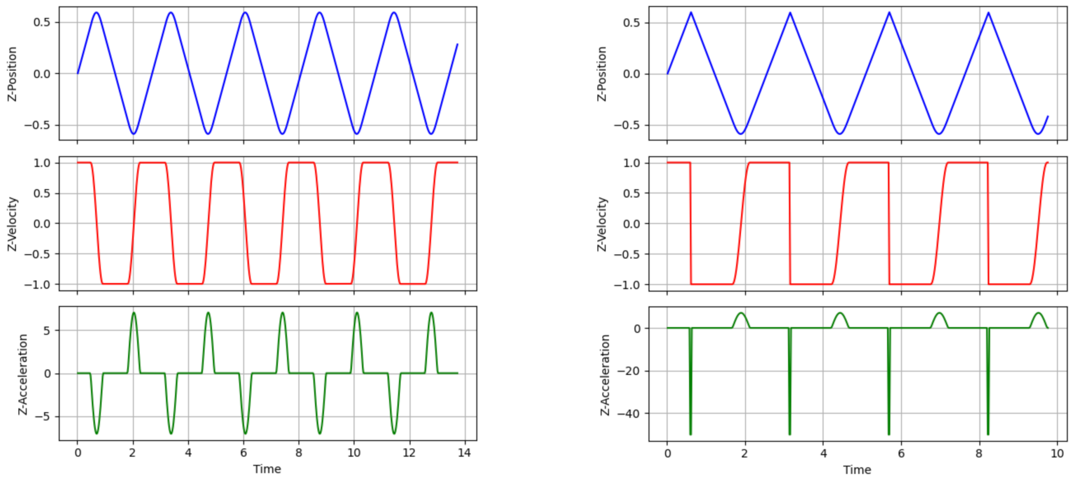
For this task, the right spring is removed and replaced by an elastic collision model.

```
void FixedUpdate()
{
 rb.velocity = new Vector3(0, 0, -rb.velocity.z);
}
```

This directly reverses the velocity of the car upon impact, simulating a perfectly elastic collision with a fixed object.

2.1.1 Plots and Interpretation

We created a comparison plot showing both the spring-based model and the collision-based model. The car's motion in both cases is visually similar, but with the collision model, the velocity reversal is instantaneous, while with the spring model, it's more gradual due to force-based deceleration.



(A) useCollisionFormulaForRightBumper set to **false**

(B) useCollisionFormulaForRightBumper set to **true**

FIGURE 2.1: Comparison of the two models

Chapter 3

Elastic Collision with Free-Moving Left Bumper (No Friction)

The following chapter is the solution to the task ("3. Elastischer Stoss, keine Reibung").

3.1 Initial Situation

In this configuration, the left bumper is set to move freely without friction. This simulates a physical scenario where two masses collide elastically.

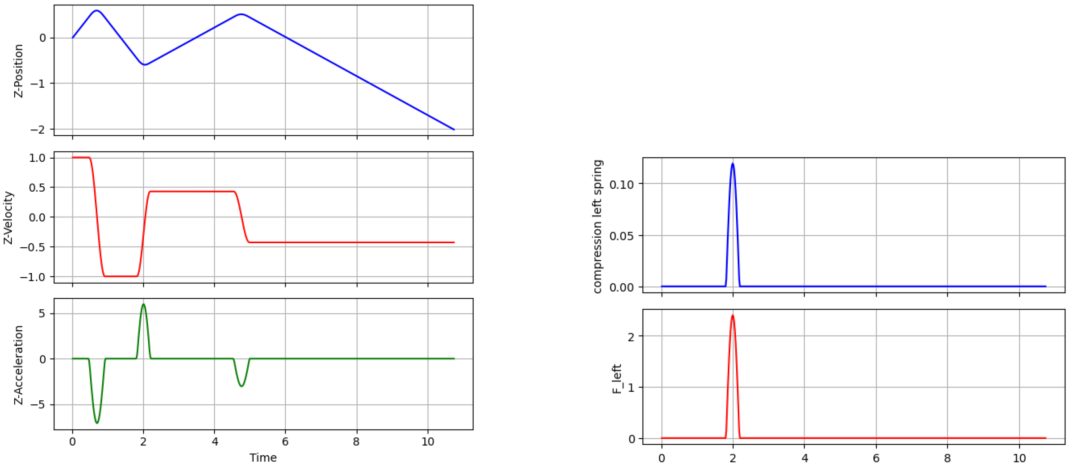
3.1.1 Implementation Changes

- The left bumper is set to **non-kinematic**.
- Movement is constrained to the **z-axis**.
- Spring force is applied to both the car and the bumper in opposite directions

This setup leads to an exchange of momentum between the two bodies during collisions.

3.1.2 Plots and Analysis

We plotted the velocities of both the car and the bumper over time. These demonstrate typical behavior of elastic collisions, where the car transfers part of its momentum to the bumper.



(A) Car

(B) Left Spring

FIGURE 3.1: Car and Left Spring plots

Our simulation adheres to these laws closely, confirming a correct implementation.

Chapter 4

Elastic Collision with Friction on Left Bumper

The following chapter is the solution to the task ("4. Elastischer Stoss mit Reibung").

4.1 Initial Situation

To simulate a more realistic scenario, we introduce velocity-dependent friction acting on the left bumper:

Friction formula:

$$F_{\text{Reib}} = -k * v$$

4.1.1 Code Implementation

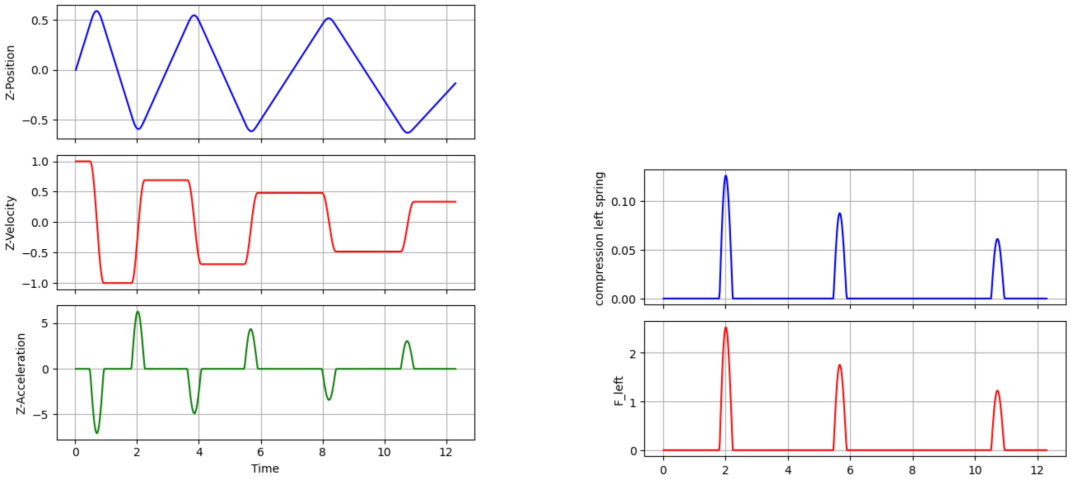
```
float frictionForce = -frictionCoefficient * leftBumper.velocity.z;
leftBumper.AddForce(new Vector3(0, 0, frictionForce), ForceMode.Force);
```

The friction coefficient k was fine-tuned so that the bumper performs 2–3 noticeable movements before settling. This damping causes the bumper (and eventually the car) to come to rest.

4.1.2 Effect on Conservation Laws

- Momentum: No longer conserved due to the non-conservative friction force.
- Energy: Total energy decreases over time as kinetic energy is dissipated into heat (simulated through velocity damping).

4.1.3 Plots and Analysis



(A) Car

(B) Left Spring

FIGURE 4.1: Car and Left Spring plots

Plots show decreasing oscillation amplitudes, confirming energy loss and illustrating the damping effect.