



1. INTRODUCTION

This dissertation investigates whether joint-based suspension produces meaningfully different drifting behaviour from raycast-based suspension in a real-time vehicle physics simulation, and whether any fidelity gain is accompanied by a proportionate computational cost. Both systems were implemented from scratch in Unity using C#, sharing an identical vehicle configuration, Pacejka Magic tyre Model, drivetrain, and weight transfer system.

2. AIM & OBJECTIVES

Aim:

To determine whether joint-based suspension produces meaningfully different drifting behaviour from raycast-based suspension, and whether any fidelity gain is accompanied by a proportionate computational cost.

Objectives:

- Implement raycast-based and joint-based suspension systems from scratch in Unity.
- Ensure both systems share identical vehicle parameters, tyre model, drivetrain, and weight transfer model.
- Develop a deterministic input replay system to ensure identical driving inputs.
- Collect and analyse telemetry data across three driving scenarios.
- Measure and compare computational performance on two different hardware architectures.
- Conduct a participant evaluation to assess perceived realism and preference.

3. EXPERIMENTAL SETUP

Driving Scenarios

Telemetry was recorded at 50 Hz across three scenarios:

- Figure-8
- 90° Constant Radius Turn
- 60 km/h Handbrake Turn

Controlled Testing

A custom input recorder and replayer ensured identical driving inputs across all test runs. Each configuration was run three times per scenario on two machines.

Telemetry Channels (Per Wheel)

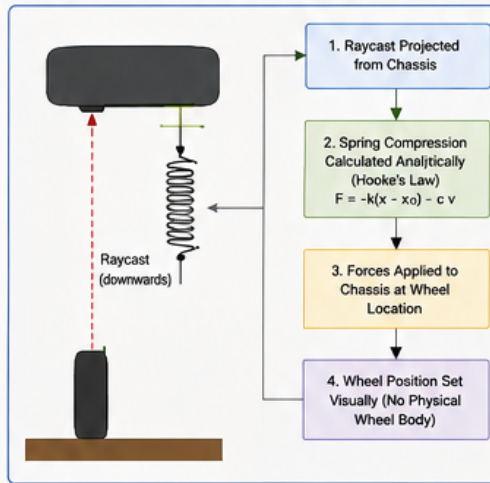
- Lateral Slip Angle (°)
- Normalised Suspension Compression (0–1)
- Suspension Compression (m)
- Suspension Velocity (m/s)
- Suspension Force (N)
- Tyre Lateral Force (N)
- Tyre Longitudinal Force (N)
- Wheel Load (N)
- Chassis Roll Angle (°)
- Physics Step Time (ms)

Key Control

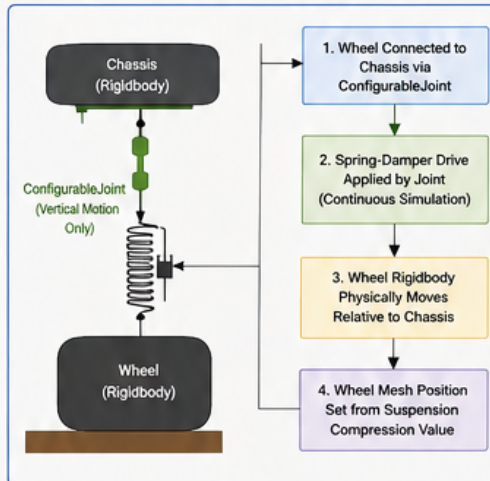
All vehicle parameters, tyre model, drivetrain, weight transfer model, and test environment are identical across both implementations.

4. SUSPENSION MODELS: METHODOLOGY COMPARISON

A. RAYCAST-BASED SUSPENSION (ANALYTICAL)



B. JOINT-BASED SUSPENSION (PHYSICAL)



5. PERFORMANCE TEST ENVIRONMENT

5.1 Performance Test Machines

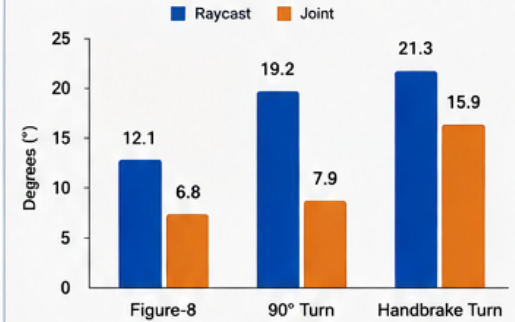
Component	Machine A (Mac)	Machine B (PC)
Machine	Apple MacBook	Personal PC
CPU	Apple M-series	Ryzen 9 5900x
RAM	16 GB unified memory	32GB DDR4
GPU	Apple M-series (integrated)	RTX 4070 Super
Operating System	macOS	Windows 10
Game Engine	Unity 6.3 (URP)	Unity 6.3 (URP)
Physics Timestep	0.02s (50 Hz)	0.02s (50 Hz)

5.2 Software Environment

Component	Specification
Game Engine	Unity 6 (version 6.3)
Render Pipeline	Universal Render Pipeline (URP)
Physics Timestep	0.02 seconds (50 Hz fixed update)
Scripting Language	C# (.NET)
IDE	Rider
Version Control	Git
Input Replay System	Custom Unity deterministic input recorder and replayer (MacroRecorder, MacroPlayer, MacroData, MacroUI)
Telemetry Logger	Custom Unity CSV telemetry writer (TelemetryWriter)

6. QUANTITATIVE RESULTS

6.1 Lateral Slip Angle (Mean, Degrees)



Raycast produced higher mean rear lateral slip angles in turning and handbrake turn scenarios.

6.2 Suspension Compression Differential (Rear Left vs Rear Right, Mean, m)

Scenario	Raycast (m)	Joint (m)	Joint / Raycast
Figure-8	0.009	0.018	2.0x
90° Turn	0.014	0.029	2.1x
Handbrake Turn	0.016	0.116	7.3x

Joint implementation produced approximately double the compression differential in turning and over seven times greater in the handbrake turn.

6.3 Computation Time per Physics Step (Mean, ms)

Machine A (Apple Silicon)

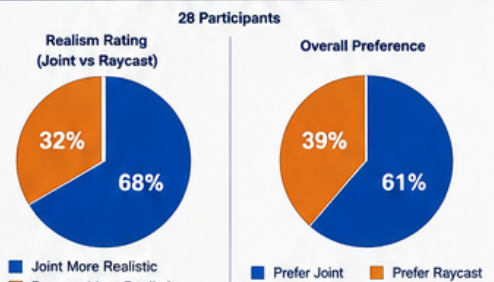
Scenario	Raycast (ms)	Joint (ms)	Joint / Raycast
Figure-8	0.18	0.42	2.3x
90° Turn	0.21	0.47	2.2x
Handbrake Turn	0.20	0.45	2.3x

Machine B (x86-64 PC)

Scenario	Raycast (ms)	Joint (ms)	Joint / Raycast
Figure-8	0.20	0.21	1.1x
90° Turn	0.22	0.23	1.0x
Handbrake Turn	0.21	0.22	1.0x

Joint model was more than double the cost on Apple Silicon but showed no meaningful difference on x86-64 PC.

7. PARTICIPANT EVALUATION



Higher mean Likert scores (1–5) across all dimensions favoured the joint implementation.

Dimension (1 = Raycast Better, 5 = Joint Better)	Mean Score
Realism	4.21
Drifting Feel	4.07
Responsiveness	3.82
Stability	4.14
Overall Preference	4.29

8. CONCLUSION

The joint-based suspension model produced meaningfully different (and in several ways more physically grounded) drifting behaviour compared to the raycast model, with a trade-off in computational cost that was architecture-dependent. Participant evaluation indicated a clear preference for the joint implementation in terms of realism and overall driving experience.

9. KEY CONTRIBUTIONS

- Two suspension systems implemented from scratch in Unity with identical vehicle configuration.
- Deterministic input replay framework ensuring fair comparison.
- Comprehensive telemetry dataset across three scenarios.
- Cross-architecture performance evaluation.
- Participant study quantifying perceived realism and preference.

10. FUTURE WORK

- Investigate optimisation strategies for joint-based suspension on ARM architectures.
- Extend testing to additional vehicle types and surfaces.
- Explore hybrid suspension approaches.
- Increase participant pool and include professional driver feedback.