

CS296 Project Report : Group 02

Shubham Jain
120050002
shubhamj@cse.iitb.ac.in

Palash Kala
120050010
palashk@cse.iitb.ac.in

Tapish Raniwal
120050023
tapish@cse.iitb.ac.in

April 10, 2014

1 Introduction

This report has been made to provide the original design and the final design of our project "Shrimp Rover : All Terrain Bot" and to discuss the differences between them. This report also includes an analysis of the submitted code as well as a discussion about its performance.

2 Design

2.1 Final Design

The original design of our project depicts a shrimp rover bot. This bot is an all terrain bot which can go through hills, mountains, roads, slope, etc. This bot had around 17 simple moving parts of which some parts are able to rotate back and forth. It consisted of a central bogey , front fork, back fork and wheels.

2.2 Final Design

The final design is almost similar to the original design. The final design has some more components as compared to the original design, as we put a spring between the two rods attaching the central bogey to the front fork. The final design is made after many trials of different measurements of friction, density and sizes of rods.

Its Spring suspension guarantees optimal ground contact of all wheels at any time and its parallel mechanism produces a passive elevation of the front wheel if an obstacle is encountered.

The bogies provide the lateral stability.

2.3 Final Design

The body of the bot is made such that centre of mass is shifted towards the front wheel which increases the climbing capacity of the bot.

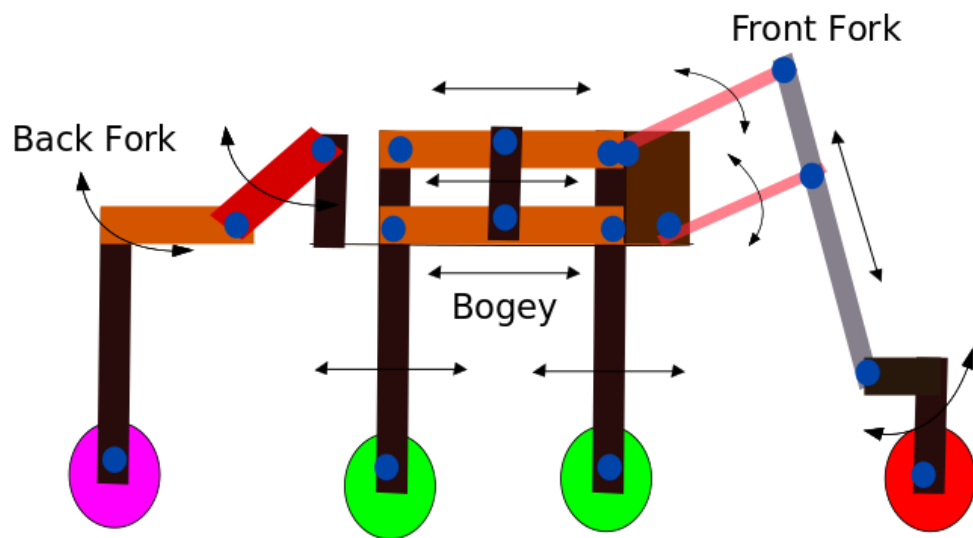


Figure 1: Shrimp Rover : Original Design

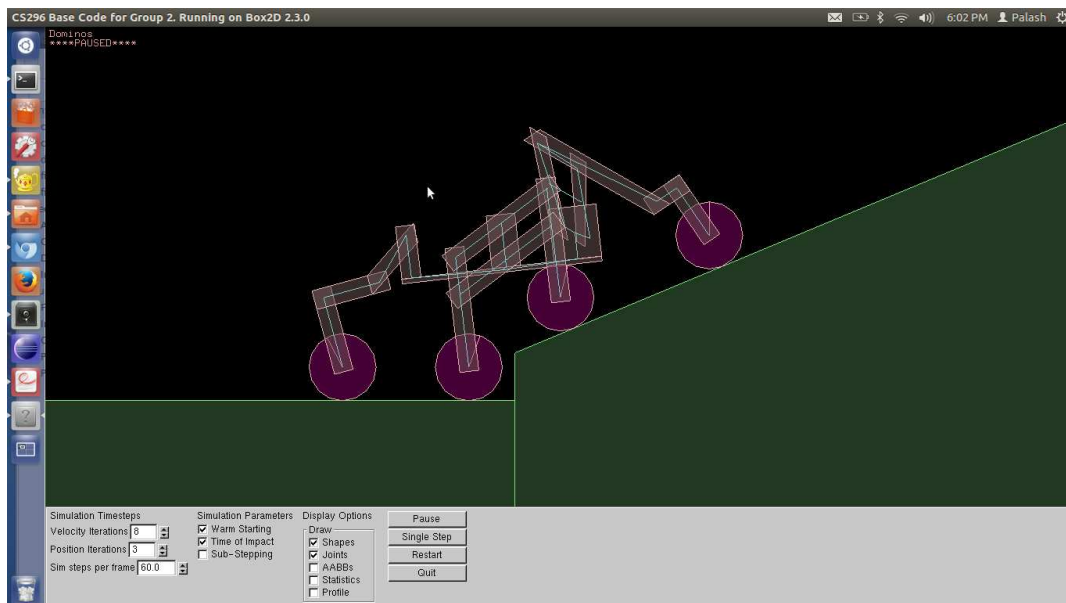


Figure 2: Shrimp Rover : Final Design

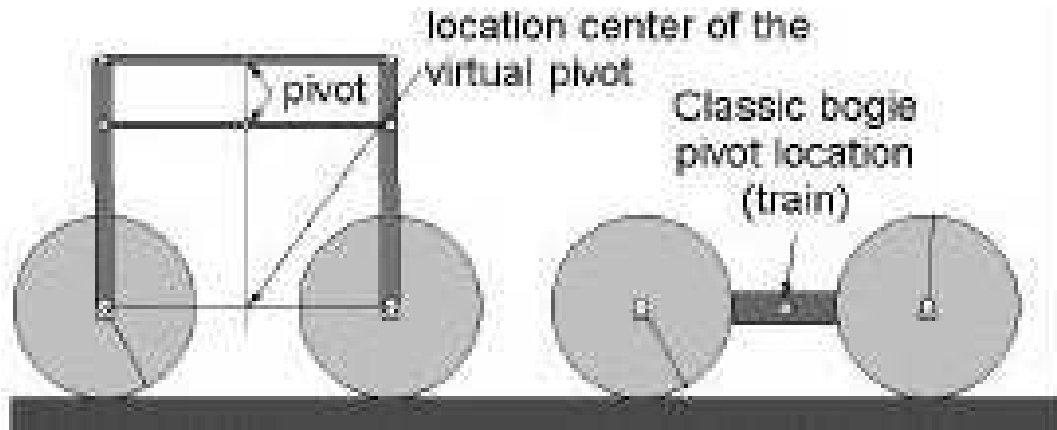


Figure 3: Shrimp Rover : Bogie

2.4 Final Design

2.5 Interesting Things

The design consists of a lot of interesting things. Even after such a simple structure, it is able to cross many terrains because of its structure and measurements. The parallel architecture of the bogies and the spring suspended fork provide a high ground clearance while keeping all 4 wheels (actually 6 wheels in physical bot) in ground-contact at any time. This ensures excellent climbing capabilities over obstacles much higher than the wheel radius and an excellent adaptation to all sorts of terrains.[1]

The body of the bot is made such that the center of mass is shifted towards the front wheel which increases the climbing capacity of the bot.

The first tyre uses the energy stored in the spring to rise up and then the second tyre rises up. After this, as the center of mass is higher

the third and fourth tyres go up easily.[2]

3 Analysis Of Code

3.1 Profiling

Release Prof O2:

Number of Iterations: 10000

Average time per step is 0.0351558 ms

Average time for collisions is 0.000746974 ms

Average time for position updates is 0.0117901 ms

Average time for velocity updates is 0.00714458 ms

Total loop time is 435.383 ms

Release Prof O3:

Number of Iterations: 10000

Average time per step is 0.0343891 ms

Average time for collisions is 0.000729776 ms

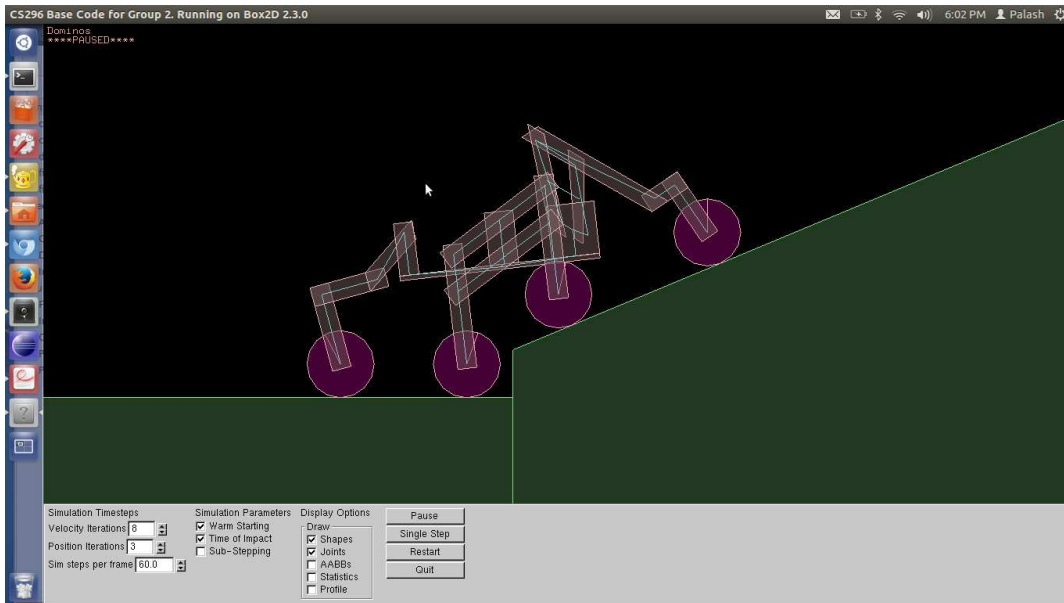


Figure 4: Shrimp Rover : Final Design

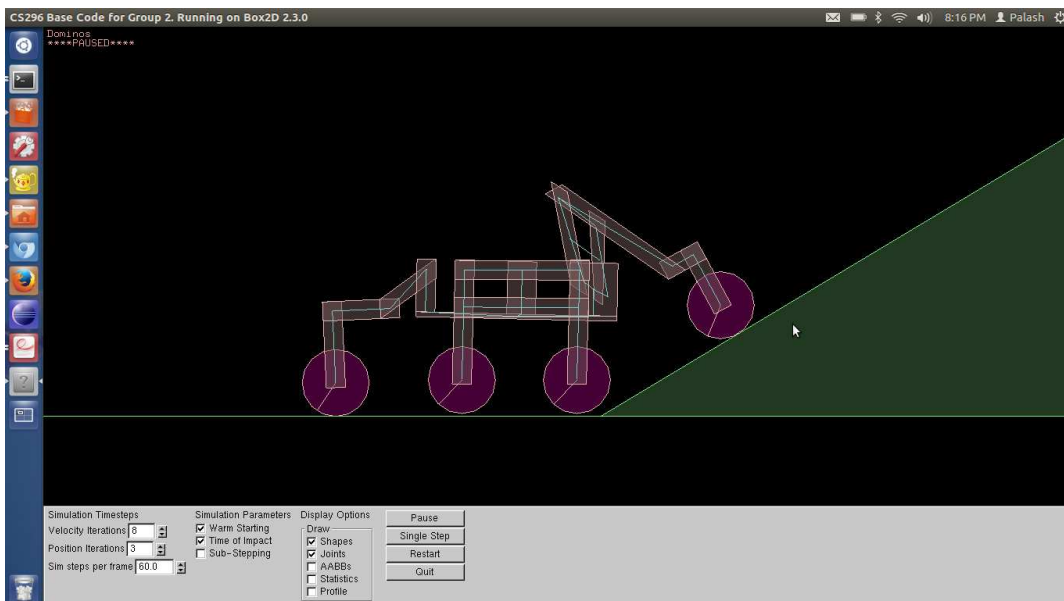


Figure 5: The energy is stored in the spring

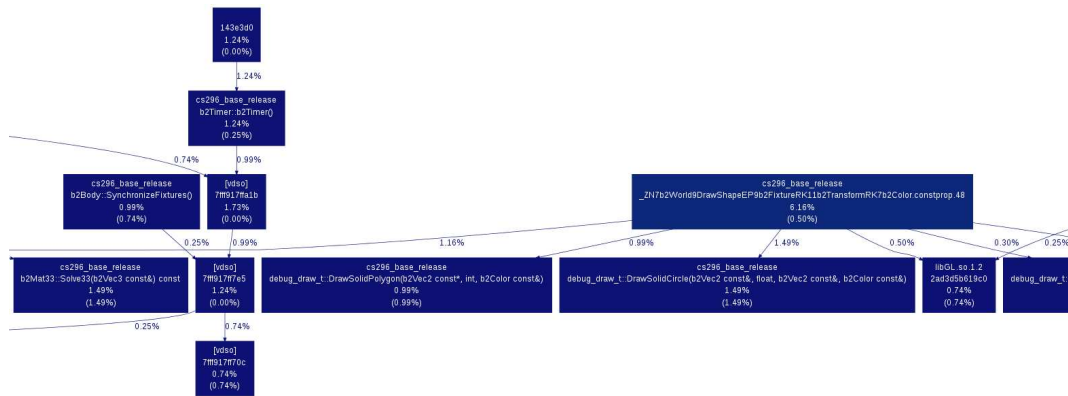


Figure 6: Call graph for O2 release

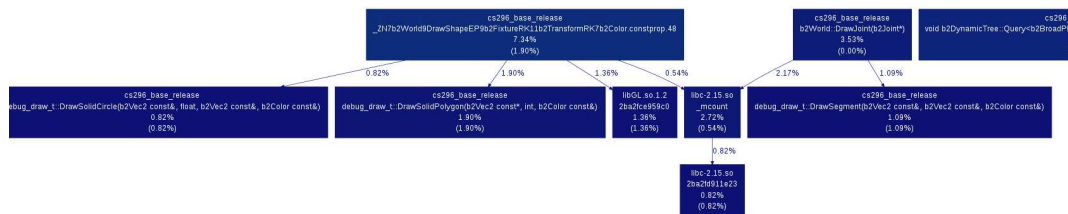


Figure 7: Call graph for O3 release

Average time for position updates is 0.0114703 ms
Average time for velocity updates is 0.00710644 ms
Total loop time is 390.961 ms

Release Prof Ofast Number of Iterations: 10000
Average time per step is 0.0352842 ms
Average time for collisions is 0.000765978 ms
Average time for position updates is 0.0117918 ms
Average time for velocity updates is 0.00718483 ms
Total loop time is 401.07 ms

Release Prof Os:
Number of Iterations: 10000
Average time per step is 0.0353292 ms
Average time for collisions is 0.000757077 ms
Average time for position updates is 0.0117607 ms
Average time for velocity updates is 0.00729658 ms
Total loop time is 450.493 ms

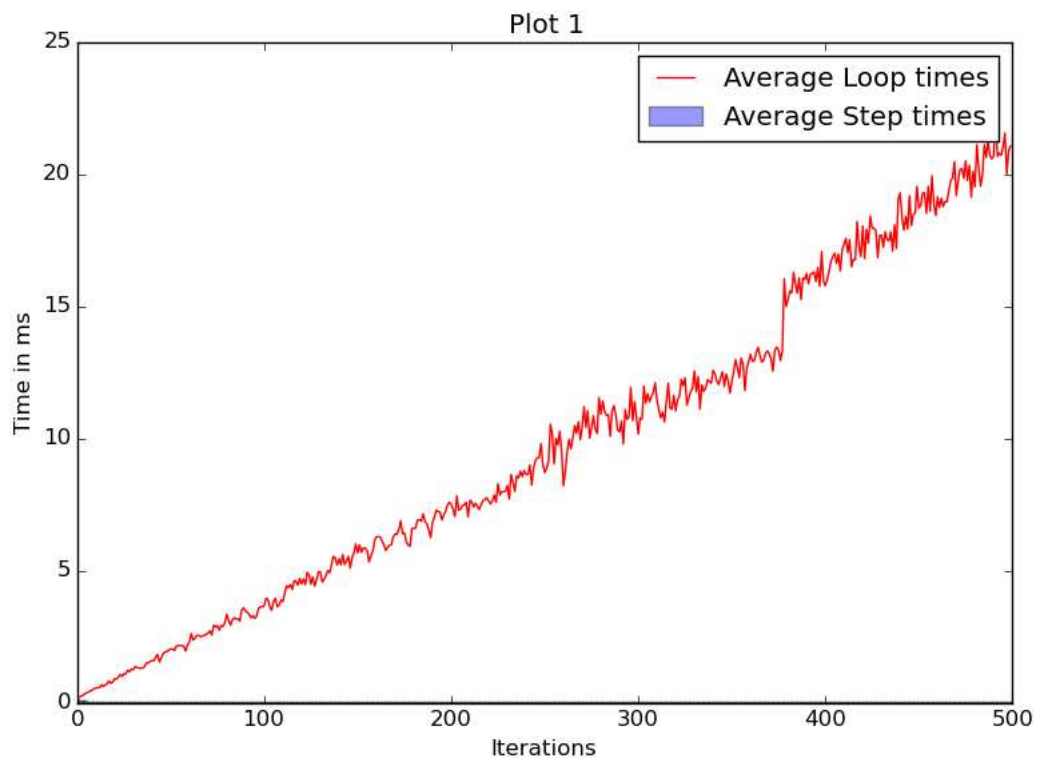


Figure 8: Figure: O3 profiling

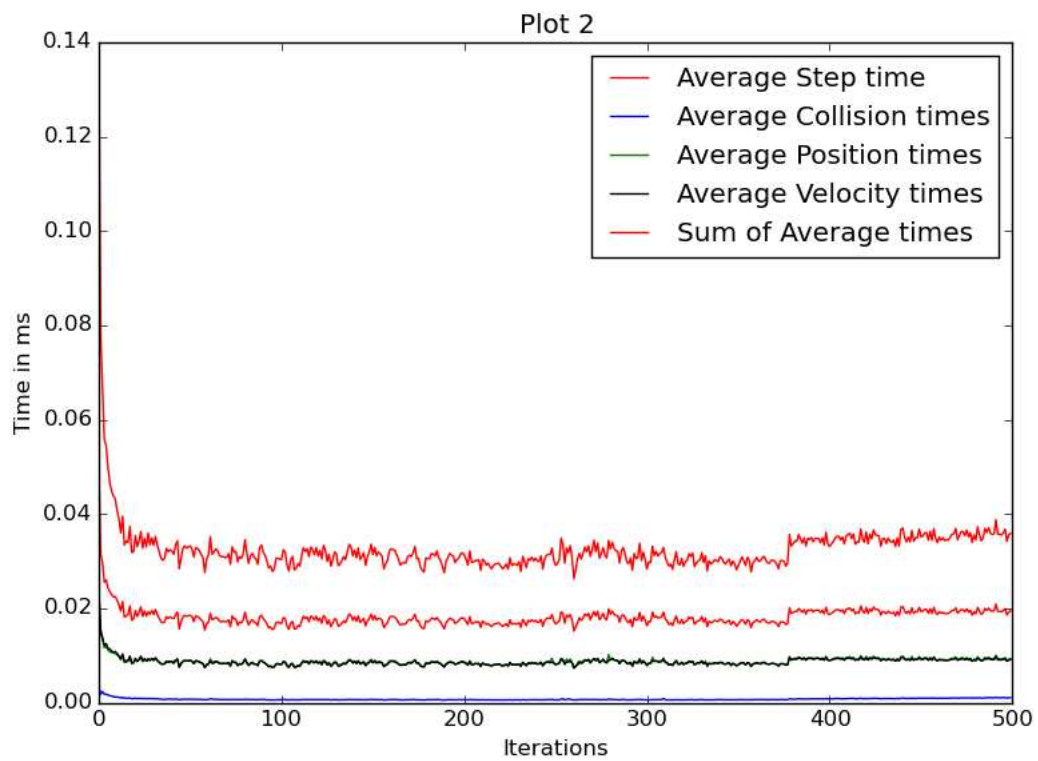


Figure 9: Figure: O3 profiling

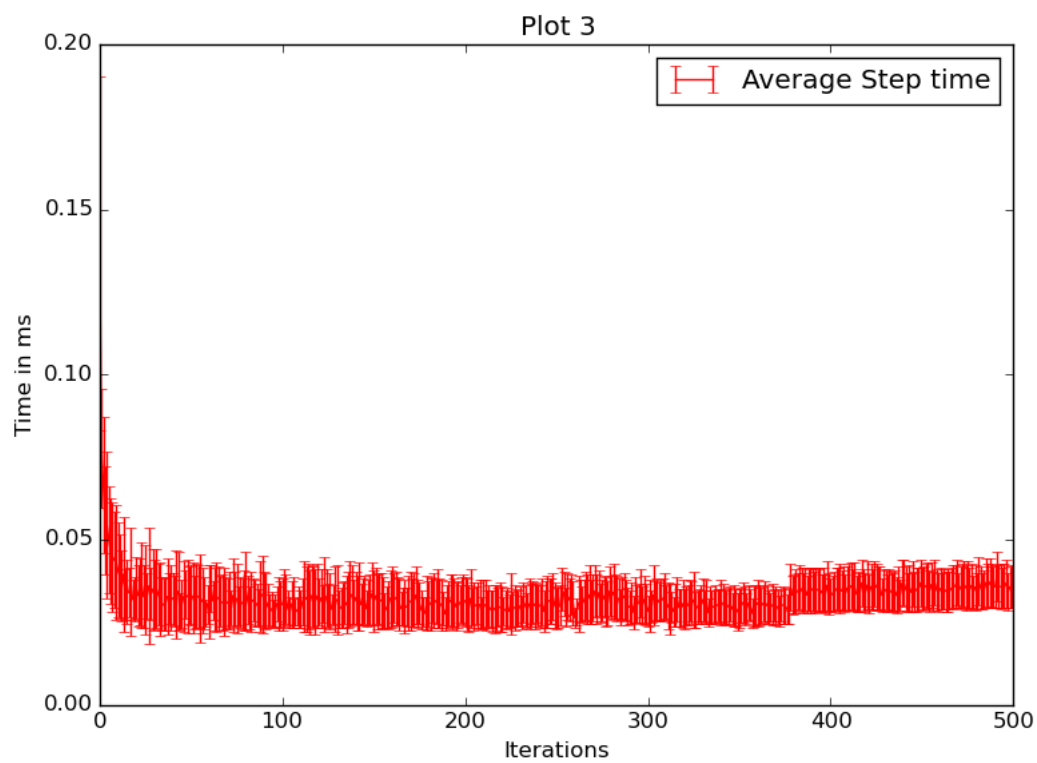


Figure 10: Figure: O3 profiling

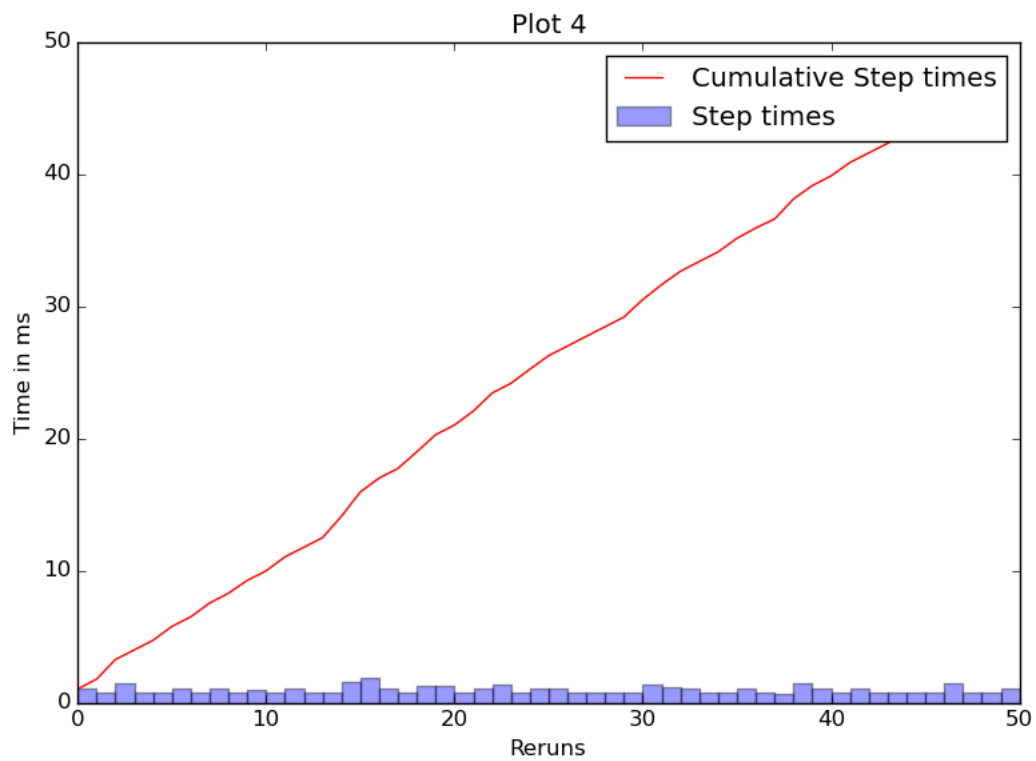


Figure 11: Figure: O3 profiling

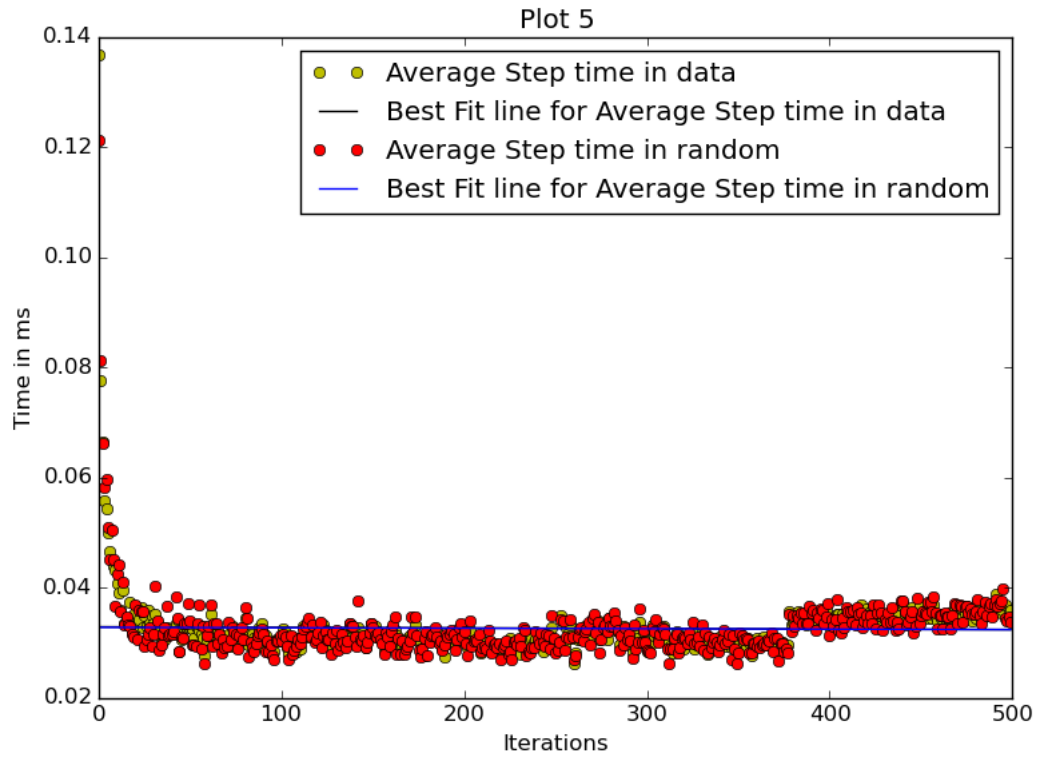


Figure 12: Figure: O3 profiling

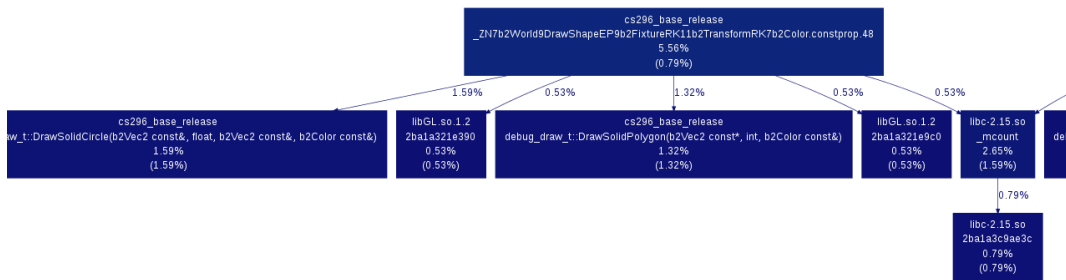


Figure 13: callgraph for release of fast

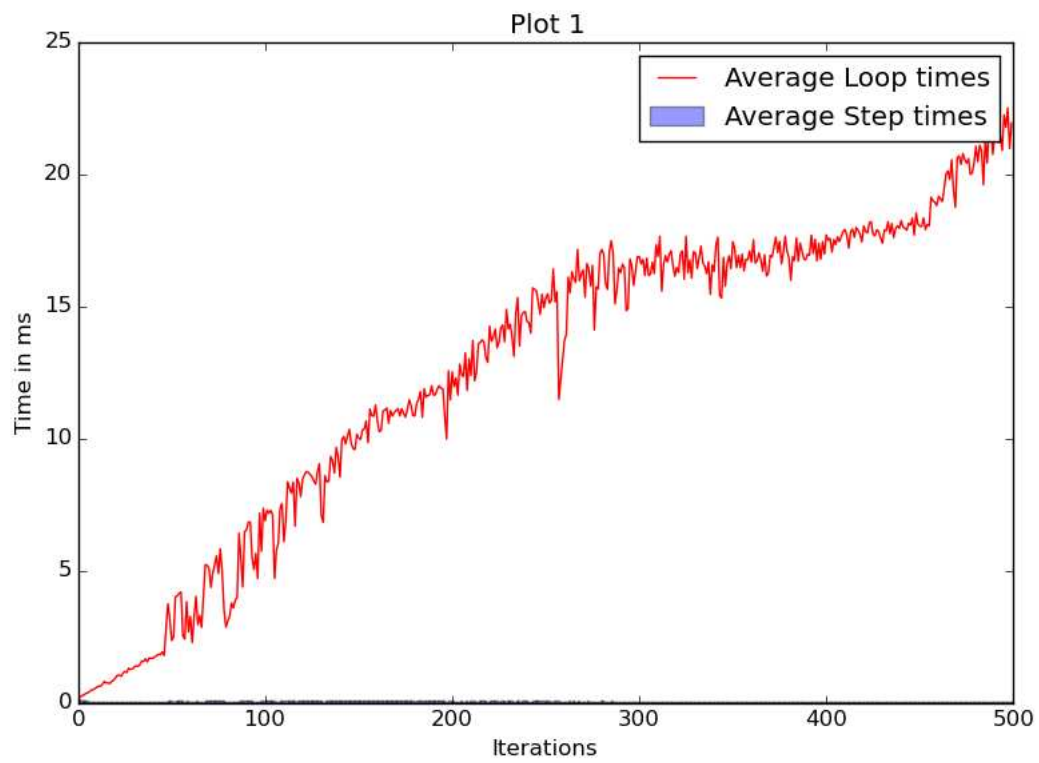


Figure 14: Figure: Ofast profiling

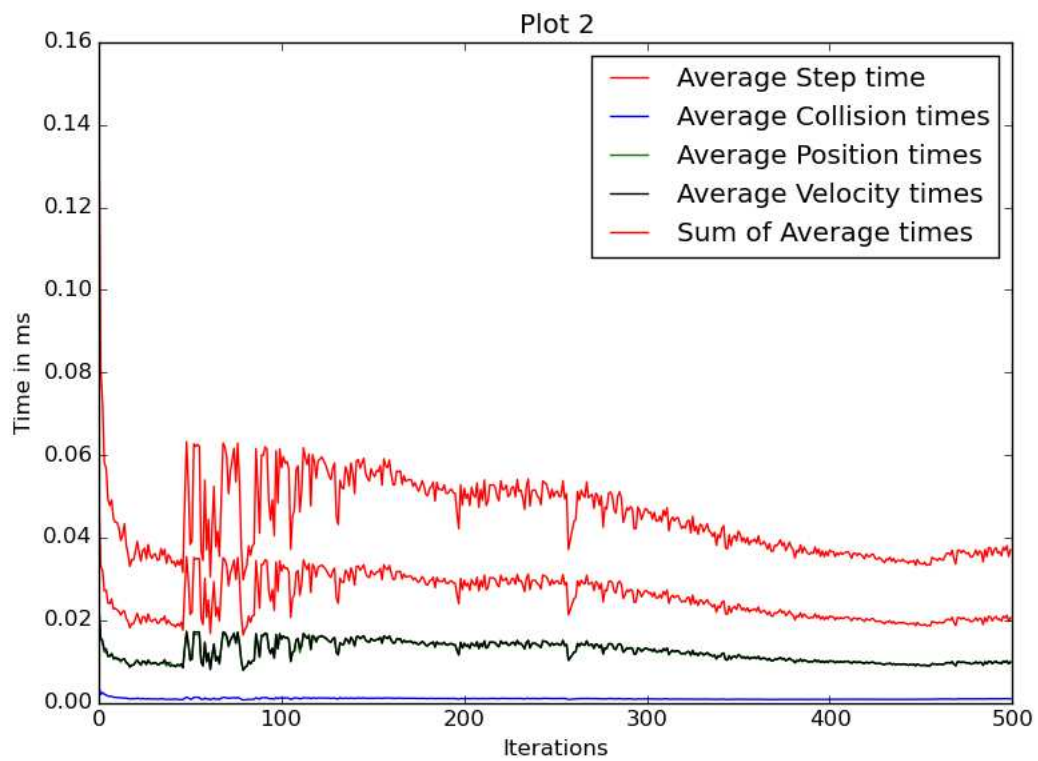


Figure 15: Figure: Ofast profiling

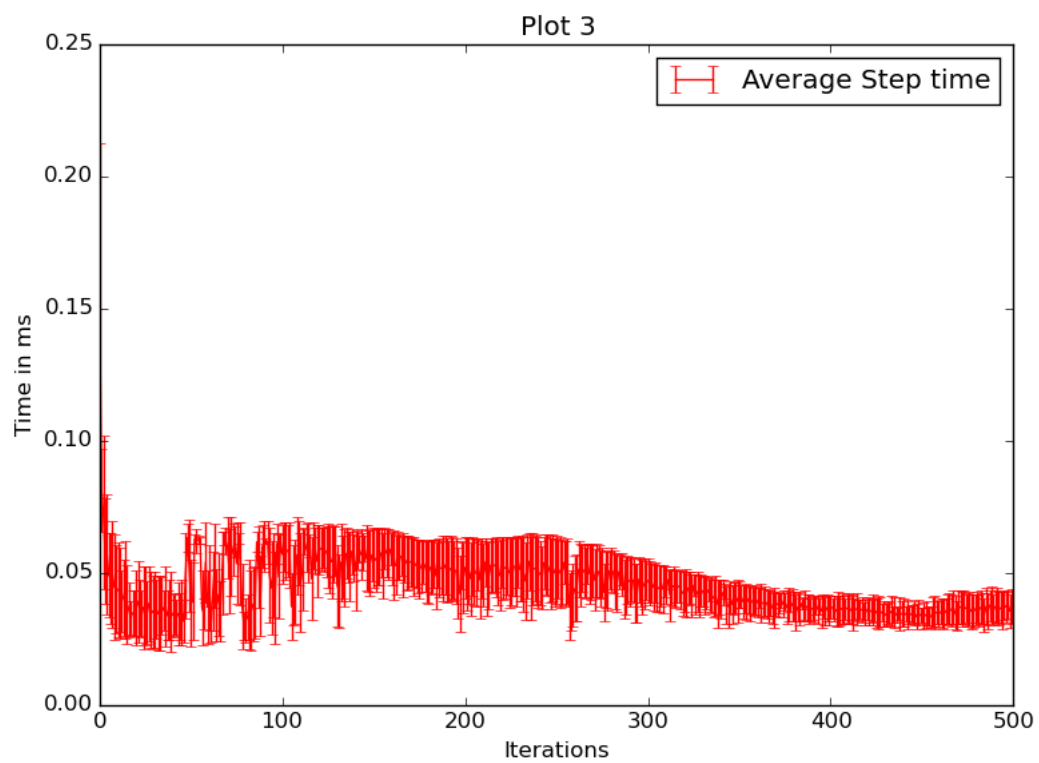


Figure 16: Figure: Ofast profiling

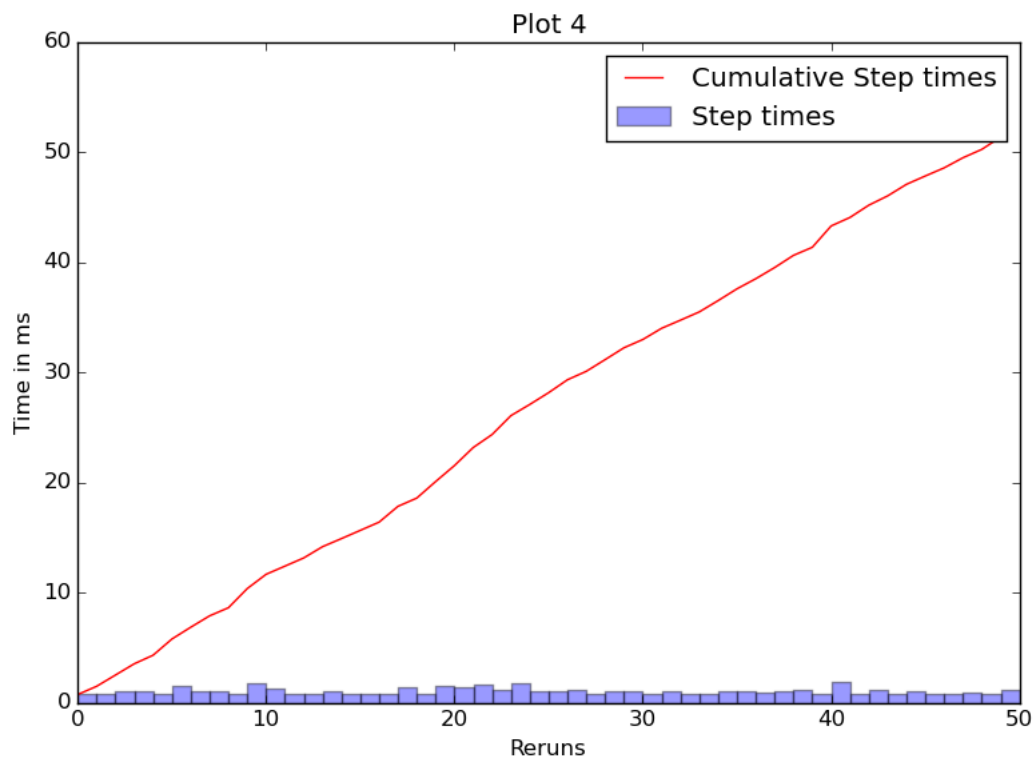


Figure 17: Figure: Ofast profiling

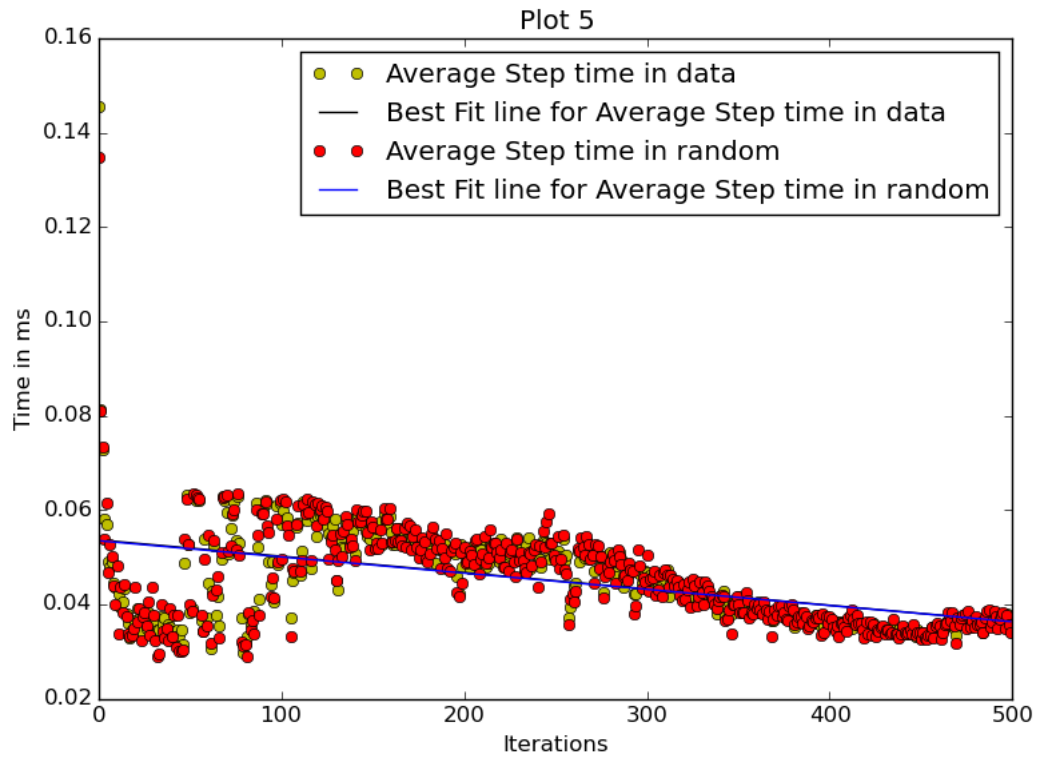


Figure 18: Figure: Ofast profiling

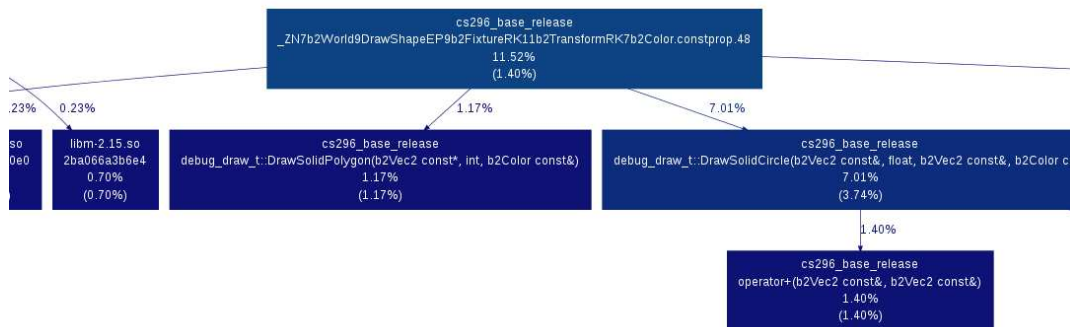


Figure 19: Call graph for release Os

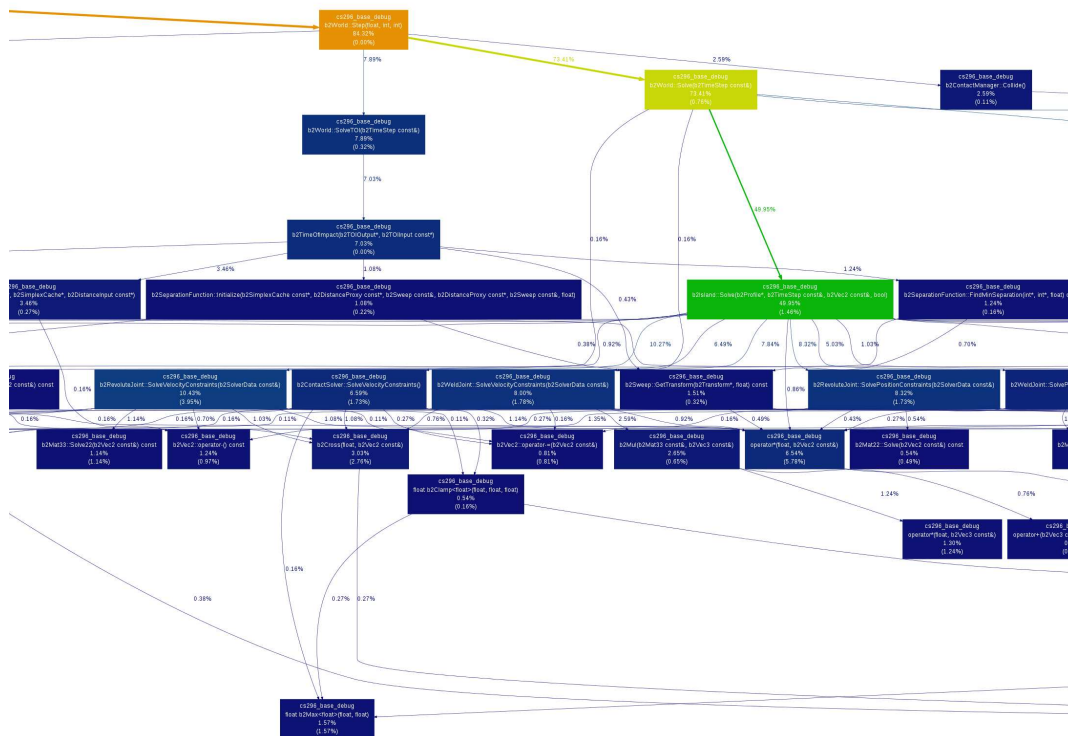


Figure 20: Call graph for debug profiling

Debug Prof:

Number of Iterations: 10000

Average time per step is 0.174029 ms

Average time for collisions is 0.00541188 ms

Average time for position updates is 0.0433578 ms

Average time for velocity updates is 0.0511959 ms

Total loop time is 1874.54 ms

3.2 Performance Optimization

After running our code for different optimizations, we found out that -O3 profiling suited best to our code.

Generally Ofast profiling works better than -O3 but for our code -Ofast profiling increased the number of cycles instead of decreasing it.

References

- [1] Yan Liang Mengyu Liang IEEE Paper by Songyi Dian, Tao Liu and Wei Zhen.
- [2] H.C. Verma. *Concepts of Physics*. Bharti Bhawan.