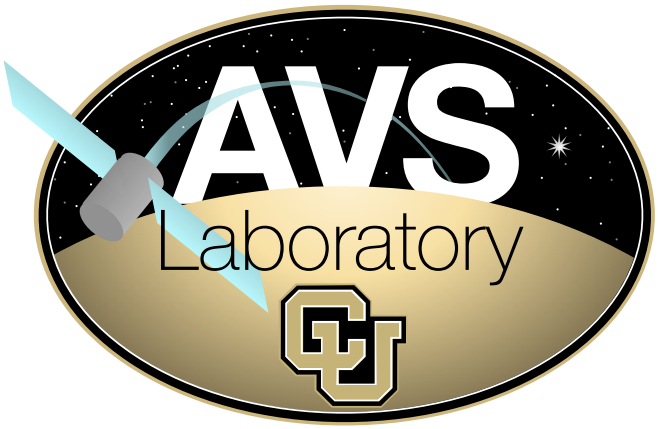

Autonomous Vehicle Simulation (AVS) Laboratory

AVS-Sim Technical Memorandum

Document ID: AVS-SIM-integrator

INTEGRATOR ARCHITECTURE IN C++ AND PYTHON

Prepared by	M. Diaz Ramos

Status: Tested
Scope/Contents
Integrator architecture model and implementation are described.

Rev:	**Change Description**	**By**
1.0	First version	M. Diaz Ramos

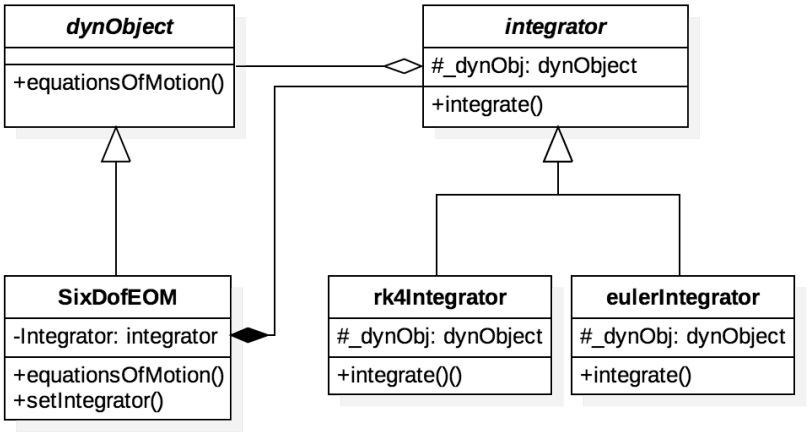

Fig. 1: UML class diagram.

Contents

---

1 Introduction

An integrator architecture model is described. How to use and create new integrators is shown.

2 Document ID

`AVS-SIM-integrator`.

3 The model

The model is shown in Figure 1. The architecture is structured around two very simple abstract classes: *integrator* and *dynModel*. The former defines the simple interface *integrate()*, while the latter defines *equationsOfMotion()*. An integrator just uses the *integrate()* method to advance one time step using the dynamics defined by *equationsOfMotion()* (also known as the F-function in the dynamics equation $\dot{X} = F(X, t)$).

In order to create a new integrator, just inherit the abstract class *integrator* and implement the *integrate()* method.

The integrator can be used by calling *SixDofEOM::setIntegrator()*.

The following snippet shows how to use an Euler integrator in Python.

```
sixDofObj = six_dof_eom.SixDofEOM()
eulerInt = six_dof_eom.eulerIntegrator(sixDofObj)
sixDofObj.setIntegrator(eulerInt)
```