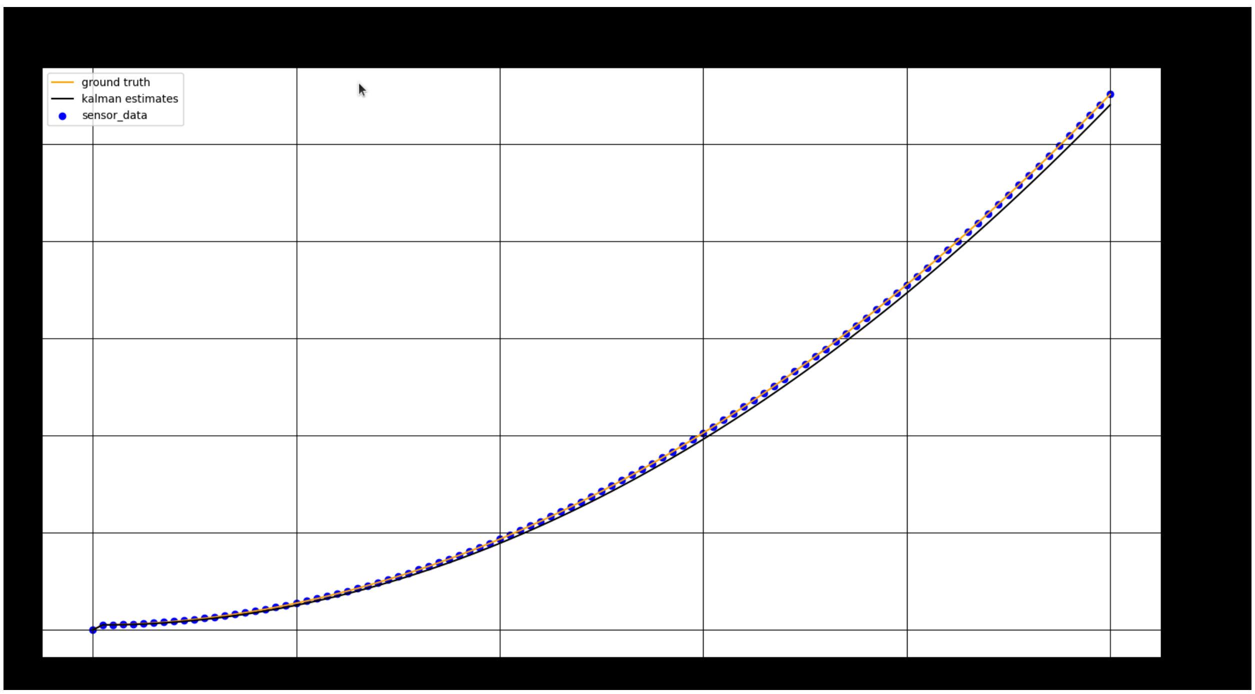
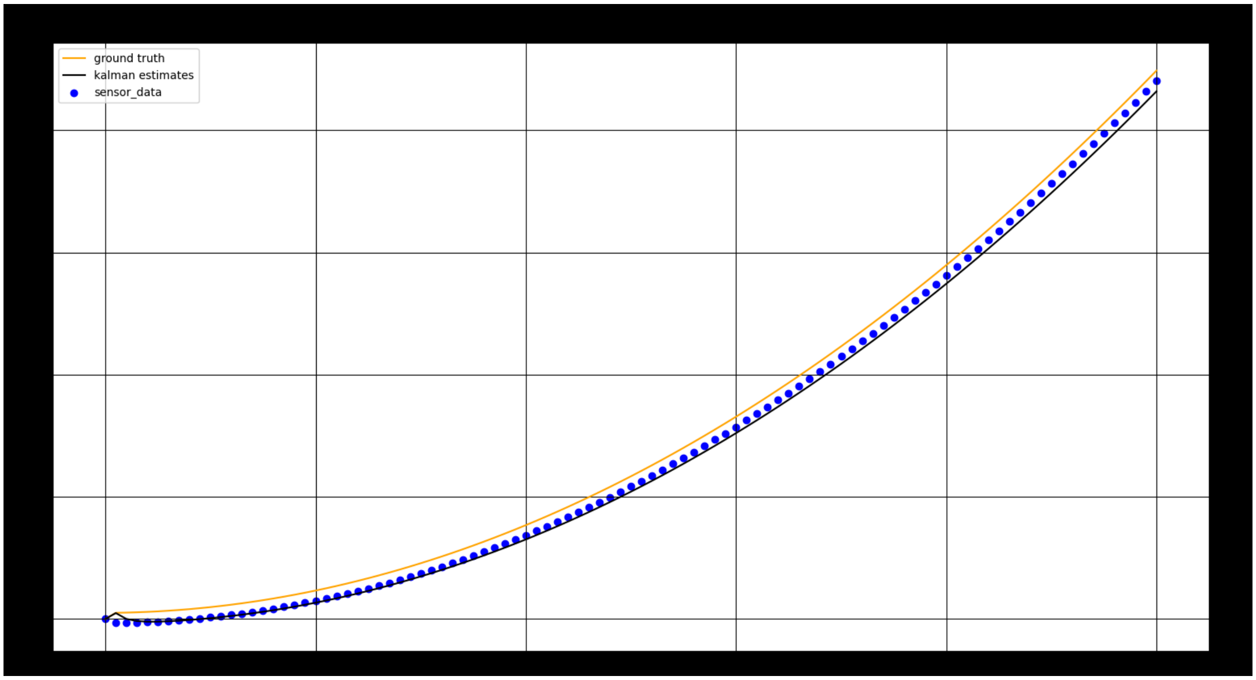
Home Assignment
Atharv Hardikar
210100034

Part 3) Following is the implementation of a simple 1-d train system given in paper by Ramsay
The implementation of KF majorly involved two stages , first was prediction where based on simple system dynamics we predicted mean state and covariance of state at time step t based on known estimate at time t-1
Implementation was done in python using np library , Initial state covariance was chosen arbitrarily and process noise covariance was determined by creating a random gaussian noise

Then having the prediction and new modeled measurement which was chosen simply as standard dynamics plus some additional gaussian measurement noise , tweaking gains of these noises I tested my KF against various scenarios results of which are shown validating the accuracy standards being met

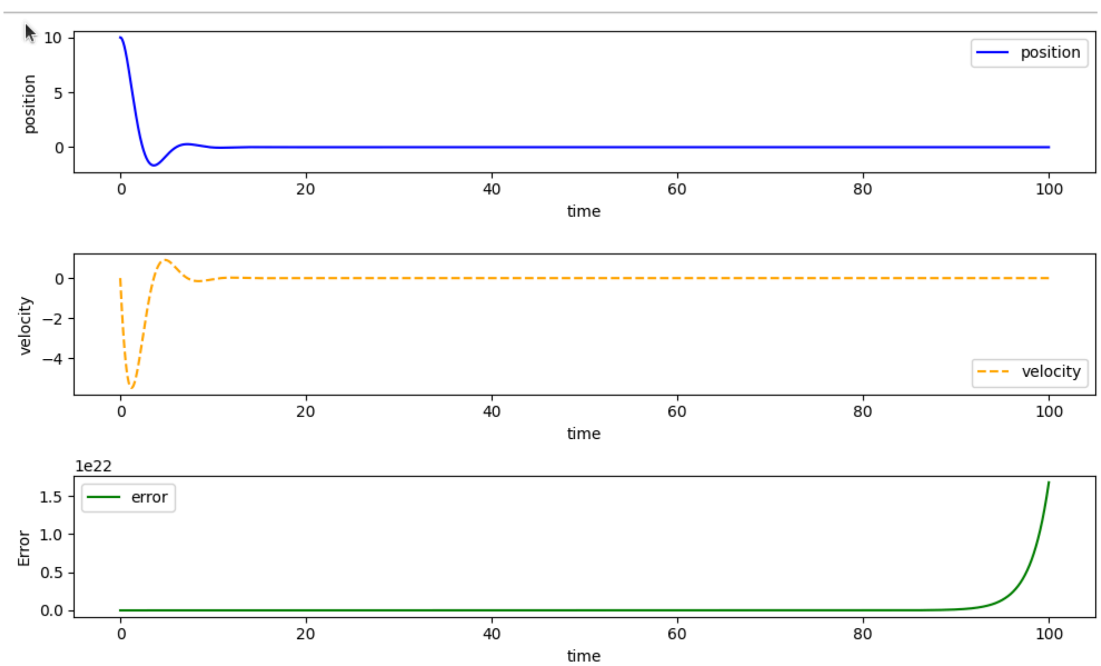


Same gaussian and process noise

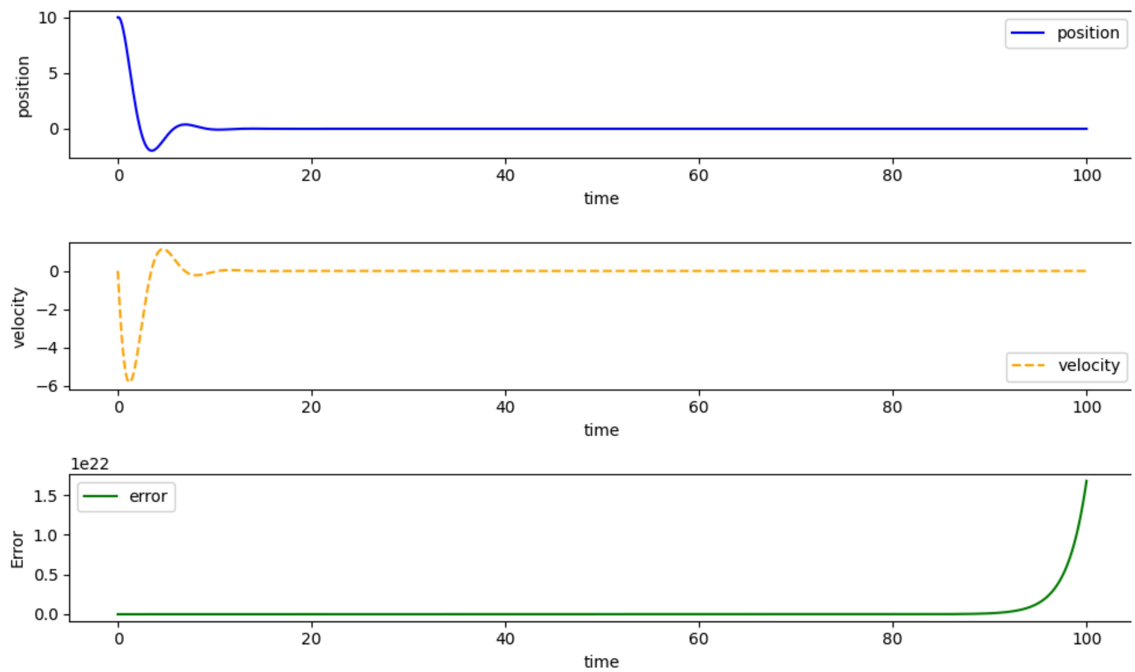


Enhanced sensor noise

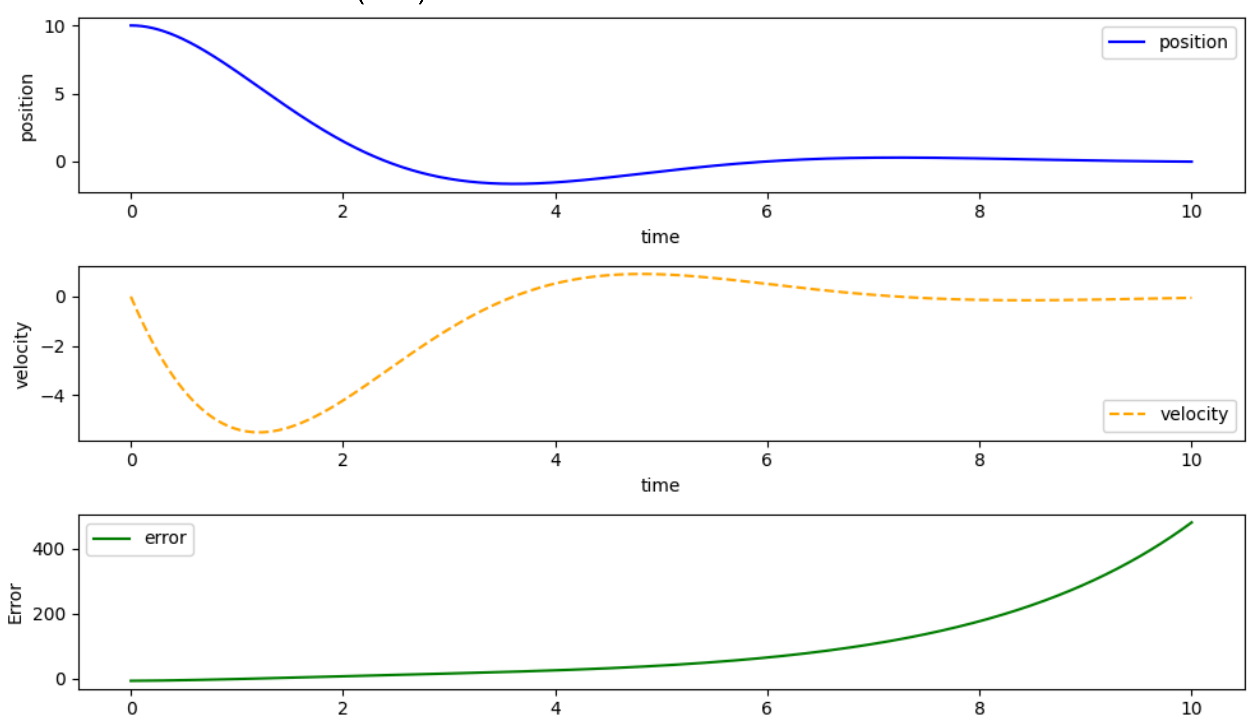
PART1) Spring mass damper (considered displaced from 10m at rest)
Integrator used : Forward Euler
Simulation time = 100 , step size of 0.01



Step size 0.1



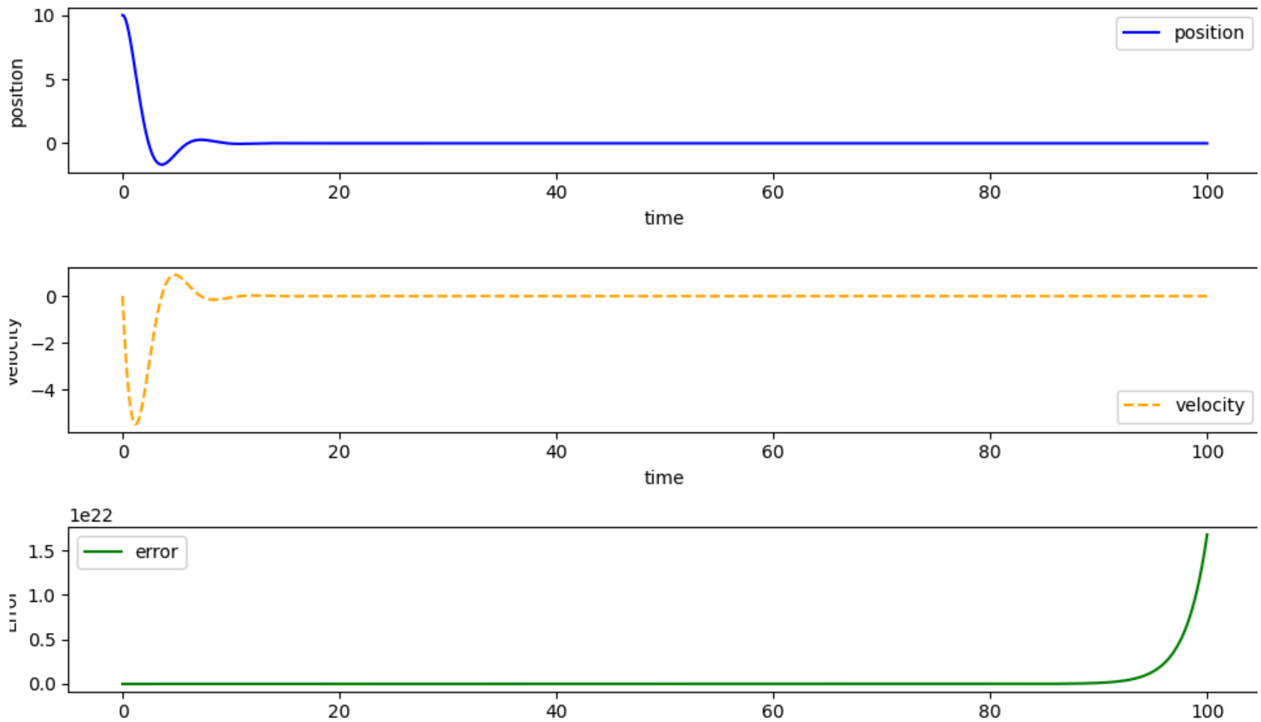
Reduced simulation time (10s)



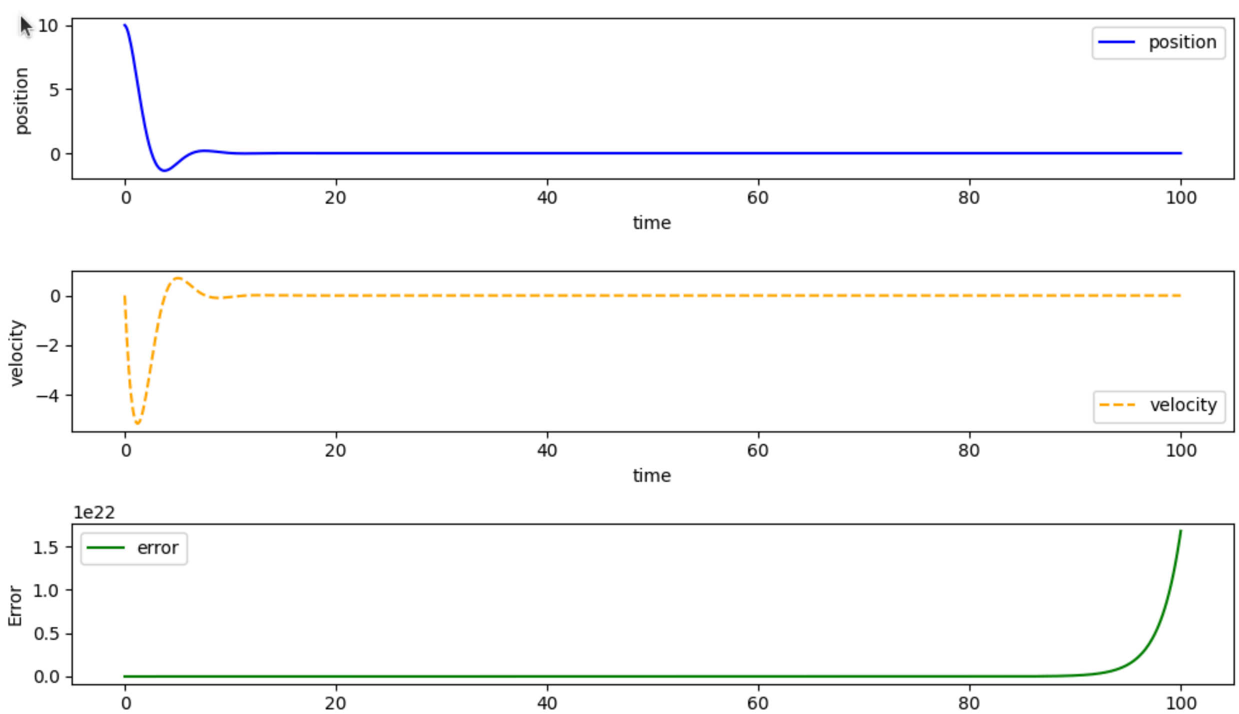
:

Repeating above iterations for Backward euler

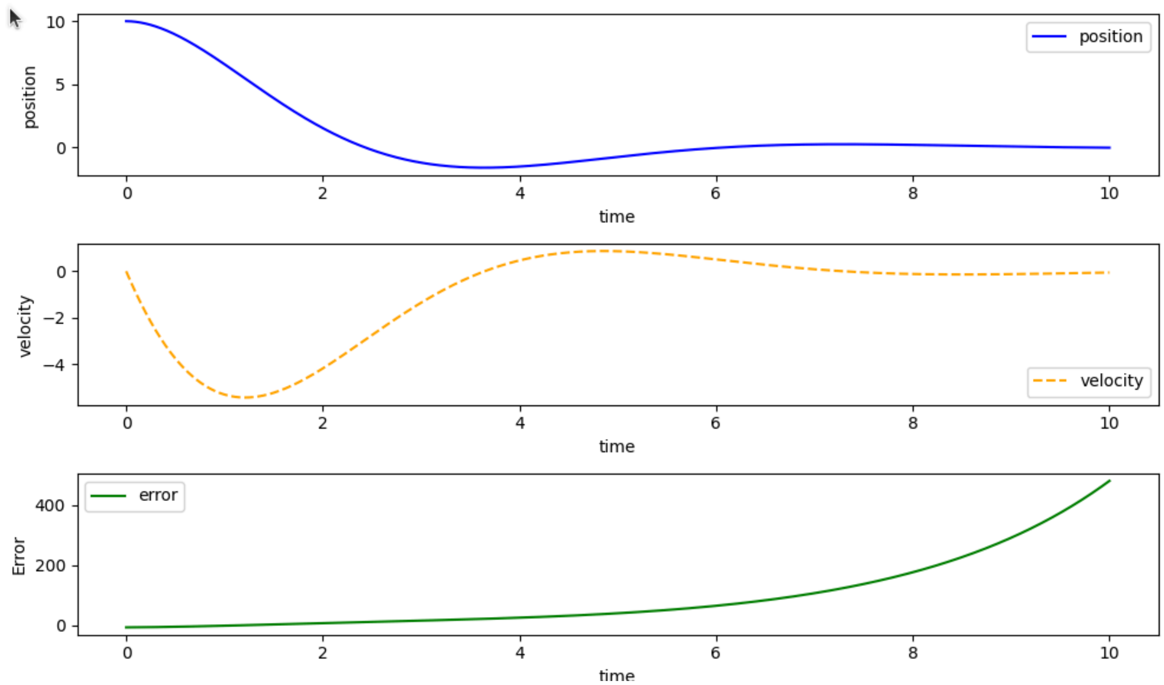
Step size = 0.01



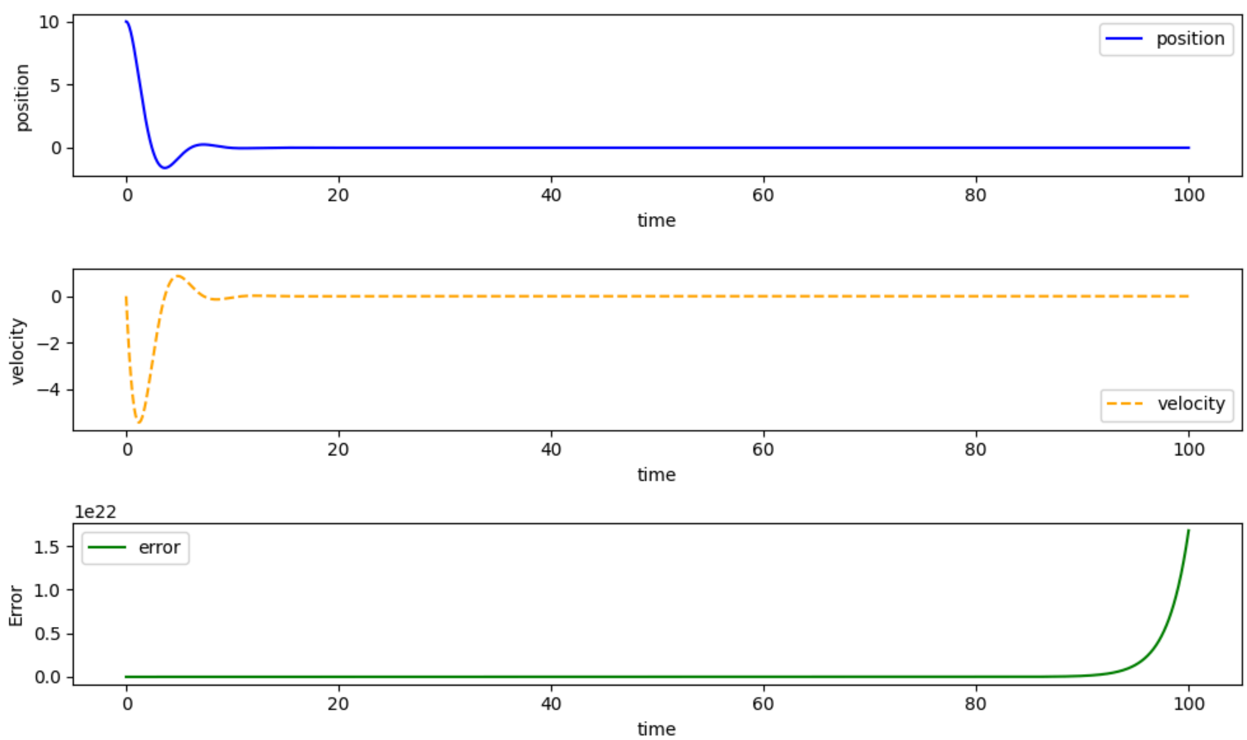
Step size = 0.1

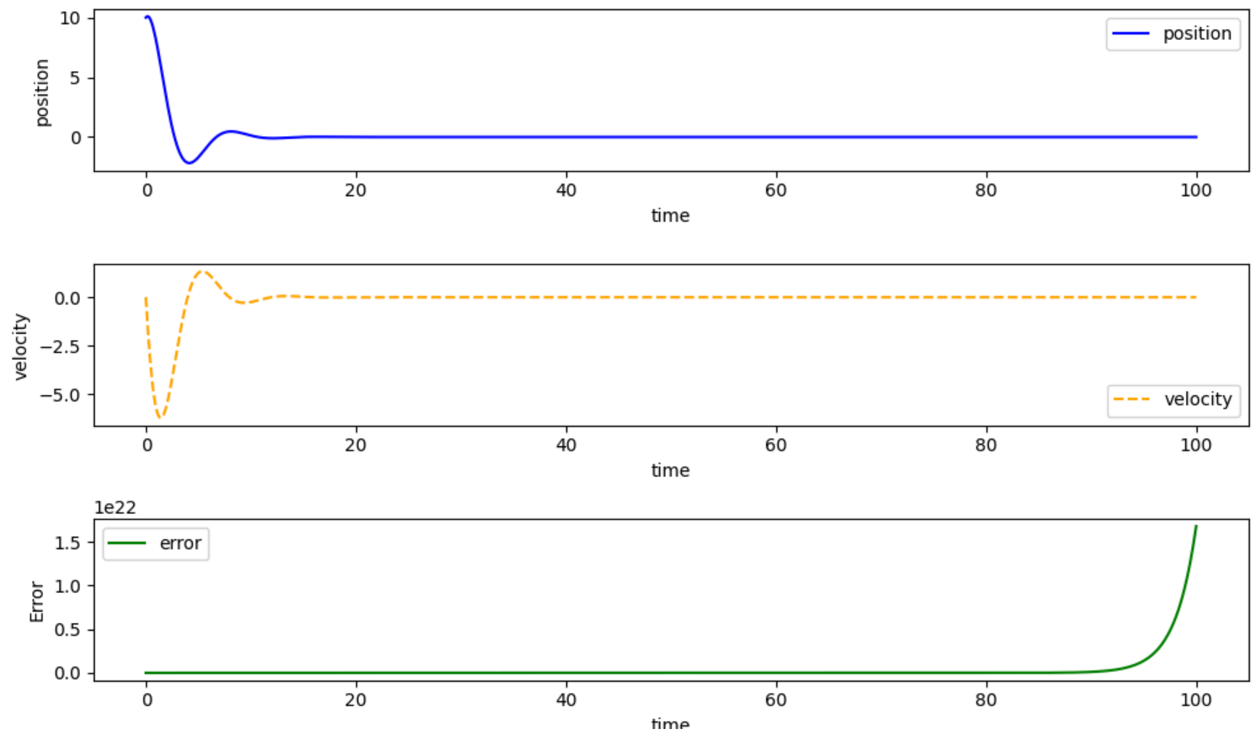


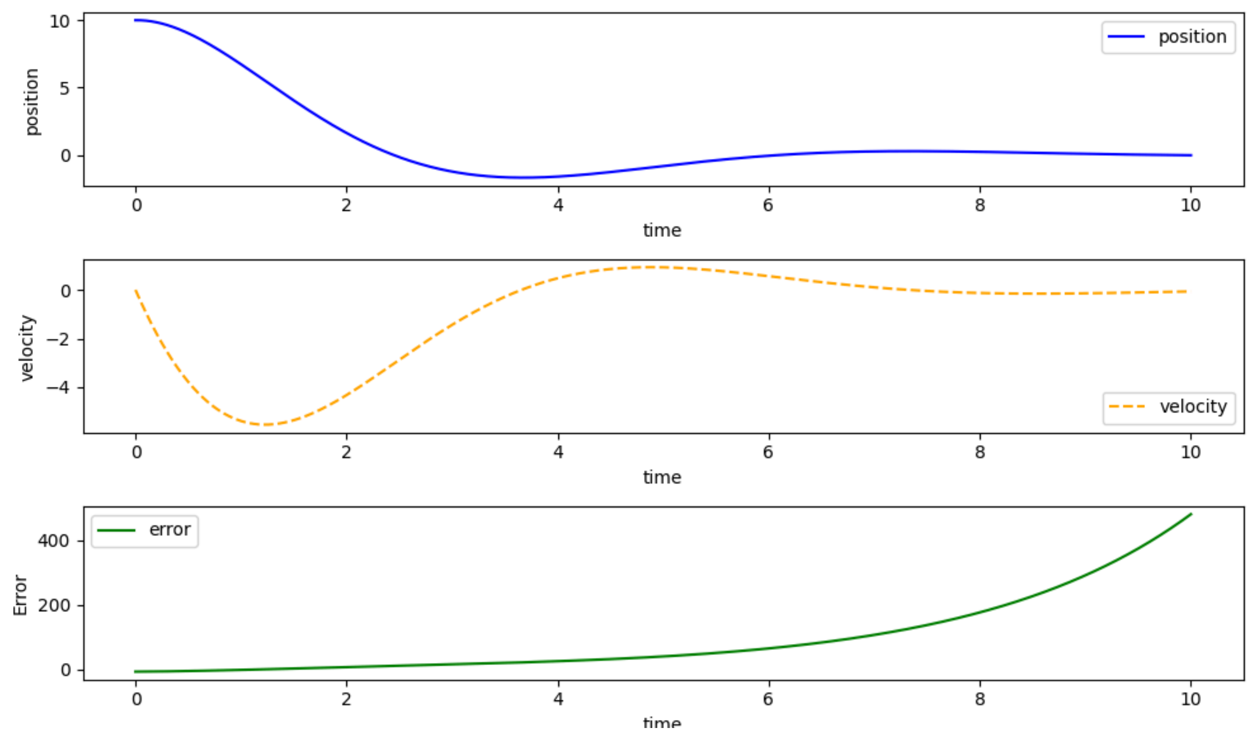
T simulation = 10s



Symplectic: implicit on 1 dimension, explicit on another dimension







Comparison:

Integrators :

Euler explicit: Prone to larger errors with smaller steps due to error accumulation but showFaster calculations compared to backward Euler.
Euler implicit: More stable for smaller steps, reduces error growth. However Introduces phase lag compared to forward Euler.
Symplectic methods: Preserve energy ,better over long simulations but Larger step sizes needed for good accuracy, potentially less efficient.

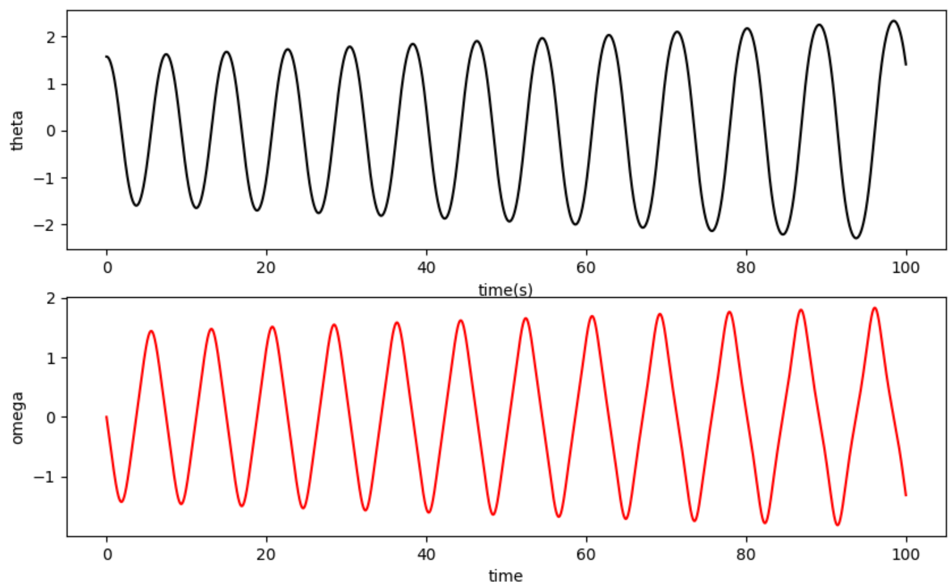
Difference however seen is negligible in our case because of simple linear dynamics

Effect of step size : due to propagation of error / numerical approximations, error increases as step size increases

Also as the simulation time increases these truncation errors accumulate, leading to significant deviations from the true solution as seen in plots

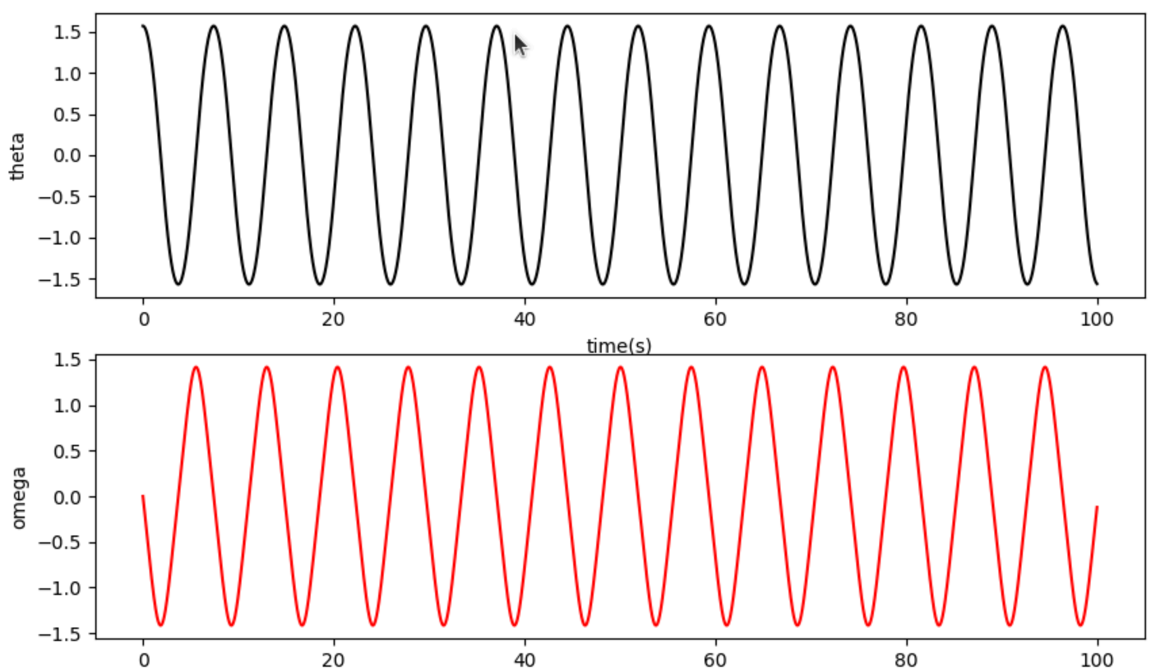
Pendulum system
Fundamental differences in terms of how theta and x live/evolve
This system is particularly more complicated to deal with in terms of numerical methods of integration because of different mathematical groups these systems are defined into
Linear space for spring mass while closed circular kind of space for pendulum
The thing is solutions just after 2pi are much closer to that just pass 0 because of circular nature , but because of propagation nature of all integrators used be it explicit or implicit imposing such constraints on theta would completely sacrifice fidelity of solutions
Also here in spring mass damper Errors in calculations gradually shrink and disappear because of the "damping" effect, which slows down the system's movements. While in pendulum error grows with time

Effect of step size and total time remains same as above
Plots for given system displaced to 90 degree and released from rest are as follows:

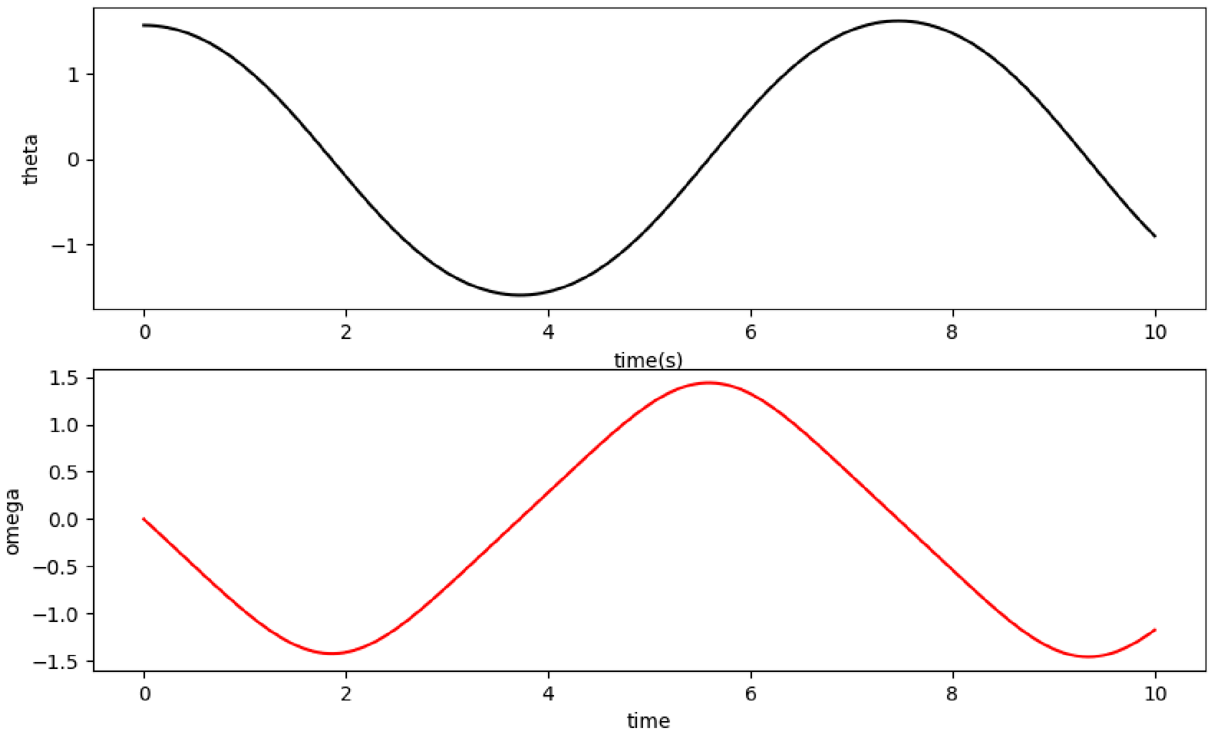


Method = forward euler, 100s simulation in step size of 0.01

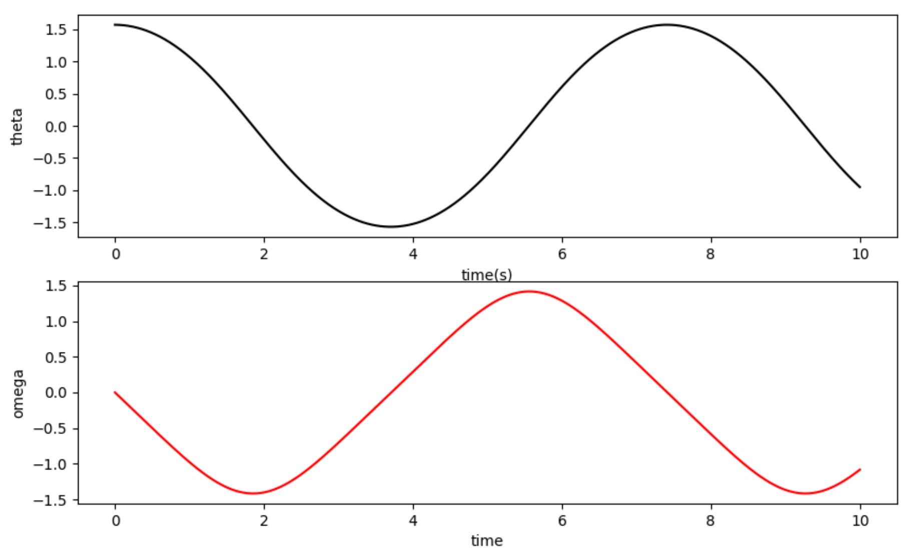
Compared to symplectic euler which is more accurate in this case for its spirit is energy conservation, the results are quite accurate



Similarly
For 10 second simulation corresponding plots



Explicit euler
And symplectic euler (assumed truth value)



Part 4) If the system is observable it is possible to fully reconstruct the system from its output measurements using something called a state observer. Observability of a state space model depends on matrices A and C
The estimator we define for should make the error exponentially converge to 0 as time -> infinity , to do so we use Output injection

The Output Injection Method involves designing the observer gain matrix
L to inject an additional term into the observer dynamics that ensures convergence of the observer states to the true system states.
The observer dynamics are given :

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

Here L is the gain matrix to be determined and y is output
Prove that given notion of estimator eventually makes the error go to 0

For a continuous-time linear system

$$\dot{x} = Ax + Bu,$$
$$y = Cx + Du,$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^m, y \in \mathbb{R}^r$, the observer looks similar to discrete-time case described above:

$$\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - \hat{y}\right).$$
$$\hat{y} = C\hat{x} + Du,$$

The observer error $e = x - \hat{x}$ satisfies the equation

$$\dot{e} = (A - LC)e.$$

The eigenvalues of the matrix $A - LC$ can be chosen arbitrarily by appropriate choice of the observer gain L when the pair $[A, C]$ is observable, i.e. observability condition holds. In particular, it can be made Hurwitz, so the observer error $e(t) \to 0$ when $t \to \infty$.

Source : wikipedia