```python
import numpy as np
import massParam as P

class controllerPID:

    def __init__(self):
        self.kp = P.kp
        self.ki = P.ki
        self.kd = P.kd

        self.F_e = P.F_e

        self.limit = P.F_max
        self.beta = (2*P.sigma-P.Ts)/(2*P.sigma+P.Ts)
        self.Ts = P.Ts

        self.z_d1 = 0
        self.z_dot = 0
        self.error_d1 = 0
        self.integrator = 0

    def update(self, z_r, y):
        z = y[0][0]
        err = z_r - z

        if err < 0.2:
            self.integrator = self.integrator + (P.Ts / 2) * (err + self.error_d1)

        self.z_dot = self.beta * self.z_dot + (1 - self.beta) * ((z - self.z_d1) / P.Ts)
        F_unsat = (self.kp * err) - (self.kd * self.z_dot) + P.F_e + (self.ki * self.integrator)
        F = self.saturate(F_unsat)

        # update delayed variables
        self.error_d1 = err
        self.z_d1 = z

        return F

    def saturate(self, u):
        if abs(u) > self.limit:
            u = self.limit*np.sign(u)
        return u
```