PiD.py

```python
def pid_calc(dirs, timestep, species, kfilt):
    vpic_info = get_vpic_info(dirs)
    dx = vpic_info['dx/de'] * vpic_info['di']
    dy = vpic_info['dy/de'] * vpic_info['di']
    mi_me = vpic_info['mi/me']

    pxx = kfilter(np.array(txx - (jx/rho)*px), kfilt)
    pyy = kfilter(np.array(txy - (jx/rho)*py), kfilt)
    pzz = kfilter(np.array(tzx - (jx/rho)*pz), kfilt)
    pxy = kfilter(np.array(tyy - (jy/rho)*py), kfilt)
    pxz = kfilter(np.array(tyz - (jy/rho)*pz), kfilt)
    pyz = kfilter(np.array(tzz - (jz/rho)*pz), kfilt)


    p = (1/3)*(pxx + pyy + pzz)
    if species == 'ion':
        particle_mass = mi_me
    if species == 'electron':
        particle_mass = 1
    ux = px/rho/particle_mass
    uy = py/rho/particle_mass
    uz = pz/rho/particle_mass


    dux_dx = (np.roll(ux,-1,axis=0)-np.roll(ux,1,axis=0))/(2 * dx)
    duy_dx = (np.roll(uy,-1,axis=0)-np.roll(uy,1,axis=0))/(2 * dx)
    duz_dx = (np.roll(uz,-1,axis=0)-np.roll(uz,1,axis=0))/(2 * dx)
    dux_dy = (np.roll(ux,-1,axis=1)-np.roll(ux,1,axis=1))/(2 * dy)
    duy_dy = (np.roll(uy,-1,axis=1)-np.roll(uy,1,axis=1))/(2 * dy)
    duz_dy = (np.roll(uz,-1,axis=1)-np.roll(uz,1,axis=1))/(2 * dy)

    theta = np.array(dux_dx + duy_dy)

    Dxx = np.array(dux_dx) - (1/3)*theta
    Dyy = np.array(duy_dy) - (1/3)*theta
    Dzz = 0 - (1/3)*theta
    Dxy = np.array((1/2)*(dux_dy + duy_dx))
    Dxz = np.array((1/2)*(duz_dx))
    Dyz = np.array((1/2)*(duz_dy))

    PIxx = pxx - p
    PIyy = pyy - p
    PIzz = pzz - p


    pid = -(PIxx*Dxx+PIyy*Dyy+PIzz*Dzz+ 2.*(pxy*Dxy+pxz*Dxz+pyz*Dyz))
    pid_rms = np.sqrt(np.mean(pid**2))
    pid = pid/pid_rms
    return pid
```