

Configuration :

The actual initial configuration of the end-effector

```
initial_config = np.array(  
    [0.1, -0.2, 0, 0, 0, 0.2, -1.6, 0, 0, 0, 0, 0, 0])
```

The initial configuration of the end-effector in the reference trajectory:

```
Tse_initial = np.array([[0, 0, 1, 0],  
    [0, 1, 0, 0],  
    [-1, 0, 0, 0.5],  
    [0, 0, 0, 1]])
```

The cube's initial configuration:

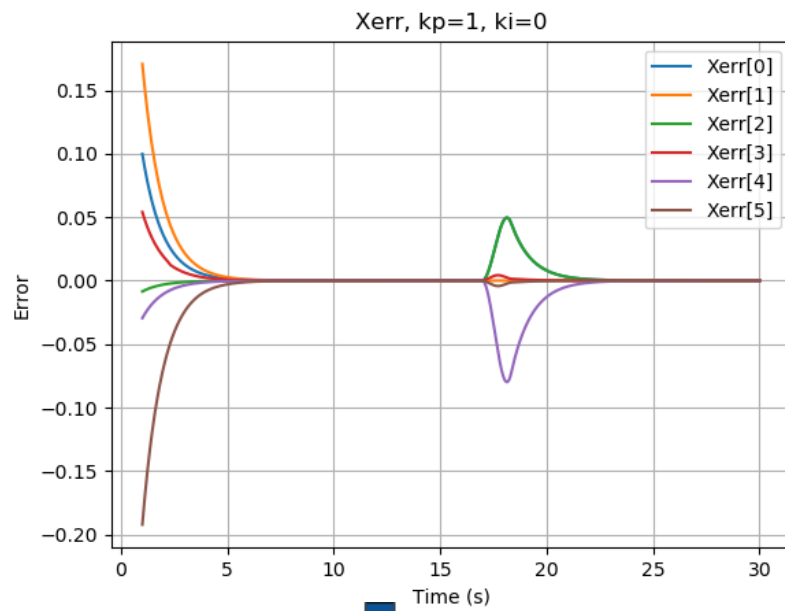
```
Tsc_initial = np.array([[1, 0, 0, 1],  
    [0, 1, 0, 0],  
    [0, 0, 1, 0.025],  
    [0, 0, 0, 1]])
```

The cube's desired final configuration:

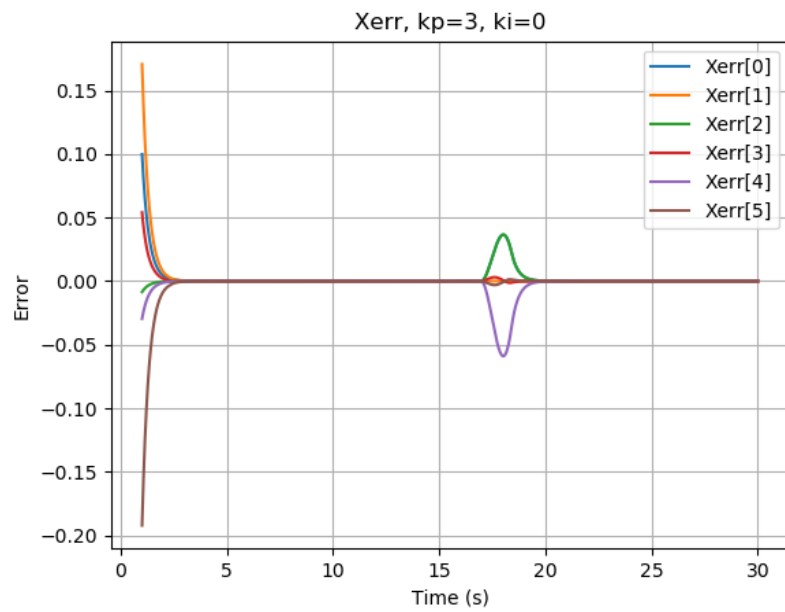
```
Tsc_goal = np.array([[0, 1, 0, 0],  
    [-1, 0, 0, -1],  
    [0, 0, 1, 0.025],  
    [0, 0, 0, 1]])
```

## **Controller Type Feedforward-Plus-PI controller**

Tuning steps

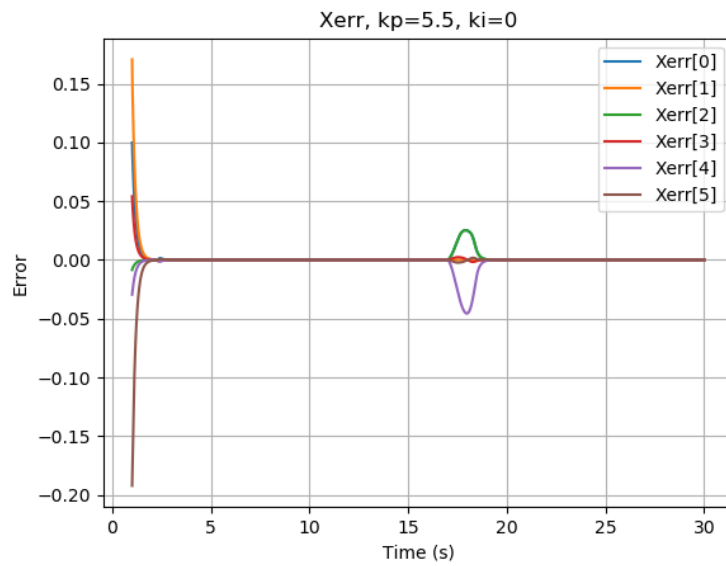


Started by setting the Integral to 0.

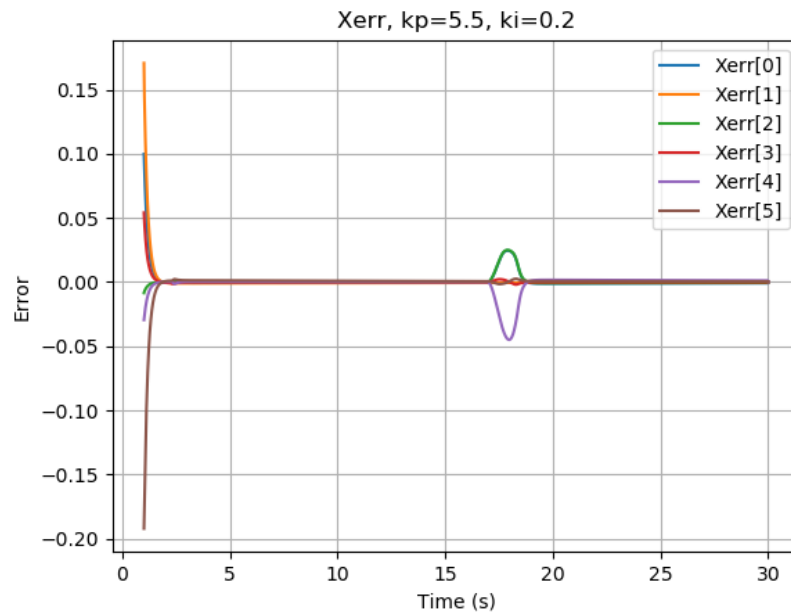


Increasing the  $K_p$  value and observed faster

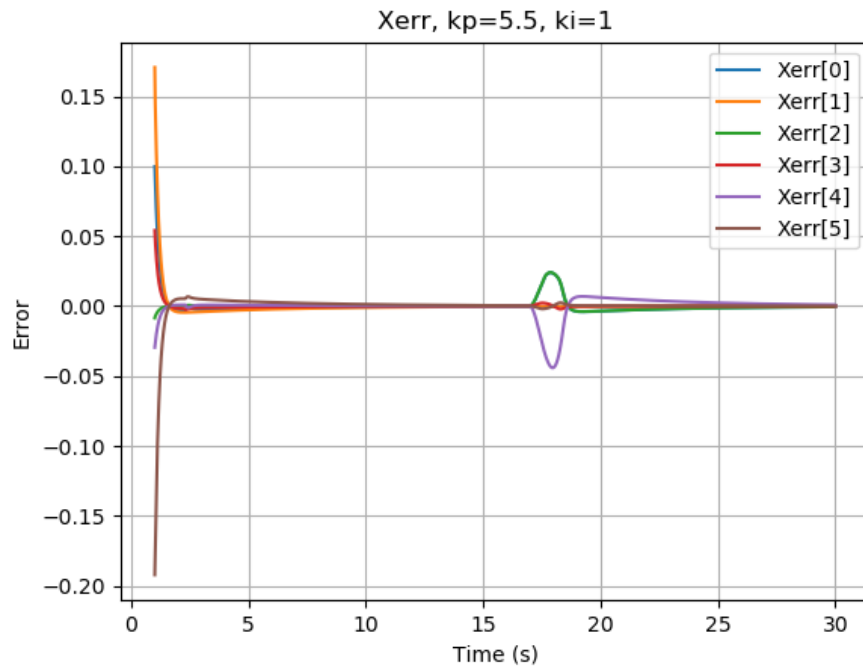
settling time



Increasing the integral gain in small increments



If  $K_i$  value becomes large convergence time right after bump and initially increase



Found Best Performance with  $K_p = 5.5$   $K_i = 0.2$

We can see that there is no overshoot, no steady-state error, and fast settling time with a little bump in between after which the plot PI again converges to 0